ESC-TR-2006-072

Project Report
CBASS-1
Revision 1

BSAR Computational Analysis and Proposed Mapping
Revision 1

M. Arakawa

5 February 2002
Reissued: 30 November 2006

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Department of the Navy under Air Force Contract FA8721-05-C-0002.

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Department of the Navy, PMS-404, under Air Force Contract FA8721-05-C-0002.

This report may be reproduced to satisfy needs of U.S. Government agencies.

The ESC Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Gary Tutungian

Gary Tutungian
Administrative Contracting Officer
Plans and Programs Directorate
Contracted Support Management

Massachusetts Institute of Technology
Lincoln Laboratory

BSAR Computational Analysis and Proposed Mapping
Revision 1

M. Arakawa
Group 102

Project Report CBASS-1, Revision 1

5 February 2002

Reissued: 30 November 2006

Lexington Massachusetts

REVISION HISTORY

Revision 1

- Incorporates performance data collected from benchmark experiments
- Uses NUWC's mapping for the first stage of beamforming
- Includes analysis of new 4K FFT algorithm mode

NOTE: This report replaces all earlier versions of Project Report CBASS-1, dated 25 September 2001, in its entirety.

EXECUTIVE SUMMARY

This project report details the MIT Lincoln Laboratory computational performance analysis of the CBASS BSAR. We describe:

- the BSAR algorithm, in both the original mode and the new, default mode
- the algorithm's computational workload for both modes
- the 12-Hammerhead DR on which the BSAR algorithm is executed
- a proposed mapping of the algorithm onto the DR
- an estimated execution time for the algorithm in both modes using the proposed mapping
- a memory usage analysis

Since the publication of the original version of this document, several key developments in the CBASS program have occurred that significantly affect the BSAR design. These developments, which will be detailed in this first revision of the report, are:

- MIT Lincoln Laboratory has completed its kernel benchmark experiments. These benchmarks, which measured the computation and communication performance of a commercially available Hammerhead board very similar to the DR, allow us to use actual measured performance instead of estimates in our execution time analysis.
- These benchmark results revealed that the throughput of the Hammerhead on small matrix multiplications is radically lower than we originally estimated. This lower computation throughput made the mapping proposed in the original document for the first step of beamforming completely ineffective. NUWC has since proposed a mapping for this step that allows us to hide the computation time behind memory access. The NUWC mapping, which allows for the use of a more computationally efficient matrix multiplication routine, also improves the execution time of this step beyond what we projected with the estimated matrix multiply throughput.
- A new algorithm mode has been adopted as the default mode. In the new, default mode, the DR performs 4K-point FFTs, instead of 8K-point FFTs, in the initial computational stage. This change, along with the change in the size of the overlap between successive data blocks, allows the time interval for each data block to exactly match the AGC loop time. Furthermore, the number of beam bands being formed has been reduced from 76 to 14. These changes significantly increase the amount of spare processing throughput (from 62% to 220%).

Our analysis indicates that the DR will be able to handle the BSAR algorithm in both the original and the new, default modes with sufficient spare processor capacity. Furthermore, there is sufficient memory for the various input, intermediate, and output data products, although some of the memory margin will be consumed in the original algorithm mode.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

The goals of the CBASS (Common Broadband Advanced Sonar System) program are centered around the upgrade of the Mk 48 heavyweight torpedo's sensors and signal processors. The Mk 48 Mod 7 torpedo will use broadband signals and signal processing techniques to improve its ability to track a target in a littoral environment and in the presence of countermeasures while maintaining its effectiveness in deep water. The torpedo's use in a littoral setting will expose it to interference from civilian shipping as well as greatly increased reverberation noise from the ocean floor and surface.

To support the use of a wider bandwidth, the Mod 7 torpedo will have a BSAR (broadband sonar analog receiver) that will perform initial filtering, decimation, and conventional beamforming of the input data from the receivers. The DR (digital receiver) portion of the BSAR will perform these operations on a custom-built embedded processor. The design of the DR is based on 12 ADSP-21160 "Hammerhead" DSPs (digital signal processors), with an aggregate peak processing throughput of 5.76 Gflop/s (billion floating-point operations per second).

MIT Lincoln Laboratory's role in this aspect of the CBASS program was to evaluate the DR and its ability to perform the BSAR signal processing in real-time. To this end, we have developed a processor assessment methodology in which we consider several candidate mappings of the signal processing algorithm onto the processor architecture and progressively refine our estimate of the processor performance.

In the original version of this report, we analyzed the computational workload associated with the BSAR algorithm, detailed several candidate mappings, and estimated the performance of the DR using each mapping. We also analyzed the sensitivity of the estimated performance to the assumptions we made about the efficiency of the DR on various computation and communication kernels. We have since measured the efficiencies of these kernels on a commercially available board whose architecture closely resembles that of the DR, and, for this revision, have incorporated these performance numbers in our analysis to generate a more accurate estimated execution time.

Another significant change since the original version of this report is the fact that a second algorithm mode for the BSAR was added. In the original algorithm mode, the BSAR collected data for 64.3 ms (with a nominal 100 kHz sampling rate) before processing was begun. This relatively long cycle time not only resulted in a slow update rate for the torpedo guidance loop, it also was not a whole multiple of the AGC (automatic gain control) loop cycle time, resulting in a mismatch between the AGC loop and the BSAR. The new algorithm mode collects data for 20.5 ms (with a nominal 100 kHz sampling rate) before beginning processing. This cycle time also meshes perfectly with the AGC loop time. The analysis in this revision will cover both the original and the new algorithm modes.

In this revision, we focus our analysis on the proposed algorithm mapping, which has been adopted by the BSAR design teams. Our analysis indicates that, for the original algorithm mode, the DR will be able to perform the BSAR signal processing in approximately 39.8 ms, which is within the 42.9 ms available, and for the new algorithm mode, the DR will be able to perform the signal processing in approximately 6.4 ms, which is within the 13.7 ms available. The other mappings, which were designed to address shortcomings of the selected mapping but were found to result in worse performance, are not considered in this revision.

2. BSAR ARCHITECTURE

The Mk 48 Mod 7 heavyweight torpedo's signal processing chain is made up of the following elements (see Figure 1):

- the pre-amplifier
- the ADCs (analog to digital converters)
- the BSAR
- the GCB (guidance and control box)

The BSAR itself consists of:

- the input FPGA (field-programmable gate array)
- the DR
- the core/control processor (not shown)
- the output FPGA

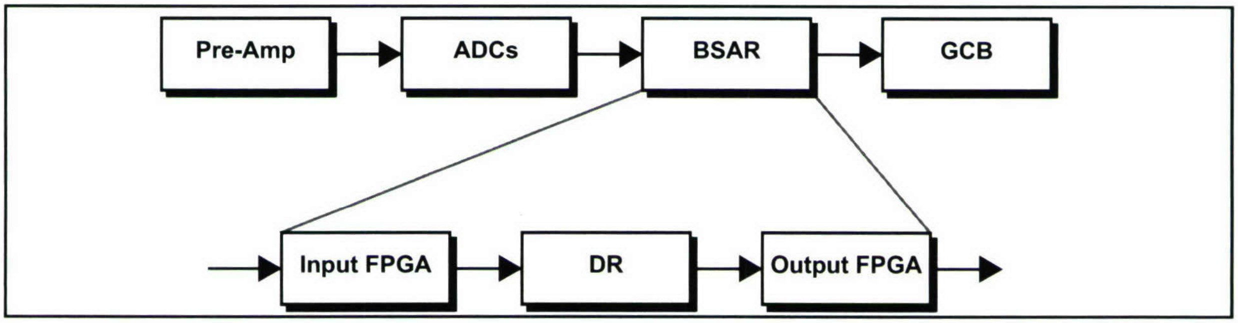

Figure 1: Mk 48 Mod 7 heavyweight torpedo signal processing chain

The BSAR is responsible for filtering, decimating, and beamforming the digitized sonar data from the ADCs. This signal processing is performed by the DR. The processed data are then sent to the GCB for further processing and detection.

The DR consists of 12 Hammerheads, each with a peak processing throughput of 480 Mflop/s (million floating-point operations per second). The 12 Hammerheads are connected to the ADCs through an input FPGA, which also converts the input data from fixed-point to floating-point representation. The DR communicates the processed data to the GCB through an output FPGA. Finally, a Texas Instruments TMS320C31 processor serves as the core processor. A block diagram of the BSAR is illustrated in Figure 2.

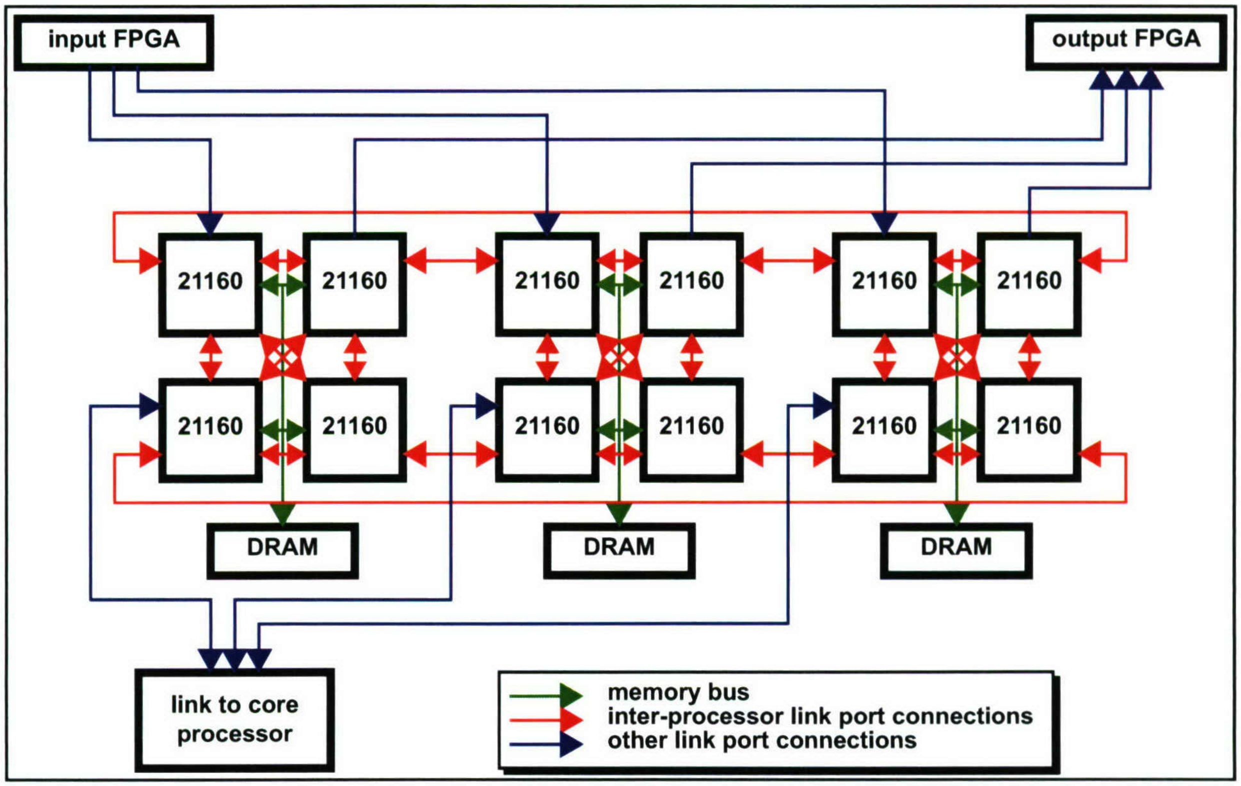

Figure 2: BSAR block diagram

The Hammerheads are organized into three clusters of four Hammerheads each. Each cluster has:

- one Hammerhead connected to the input FPGA via a link port
- another Hammerhead connected to the output FPGA via a link port
- a third Hammerhead connected to the BES (broadband evaluation system) via a link port
- the last Hammerhead connected to the interface to the core processor

The Hammerheads are connected via link ports into a two-by-six torus. In addition to the connections forming a torus, each Hammerhead is directly connected to the other three Hammerheads in its cluster via a link port (see Figure 3).

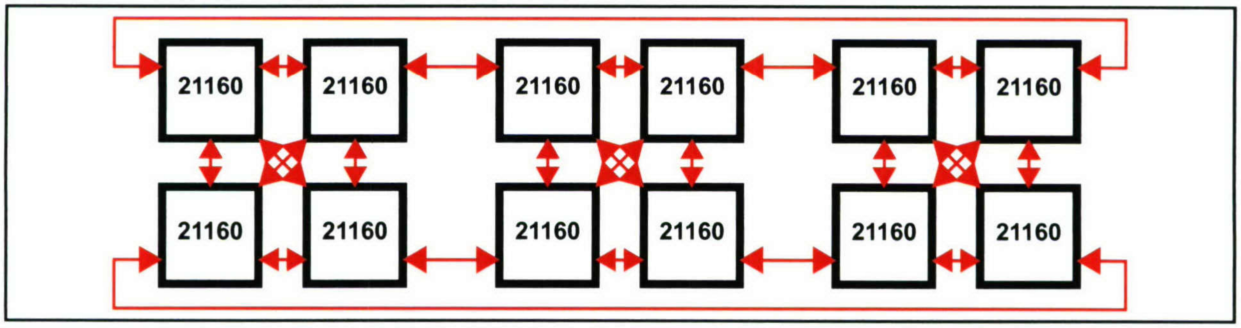

Figure 3: DR torus interconnect architecture

The four Hammerheads in a cluster share a memory bus connecting them to 34 MB (mega byte, defined as 1,000,000 bytes) of SDRAM (synchronous dynamic random-access memory)[1] and 8 MB of boot PROM (programmable read-only memory). The various elements of the DR will be described in greater detail below.

2.1 ADSP-21160 HAMMERHEAD

The Hammerhead processor consists of (see Figure 4):

- a core processor
- 524 KB (kilo byte, defined as 1,000 bytes) of on-chip SRAM[2] (static random-access memory)
- an independent I/O (input/output) processor
- an external port

[1] Given our definition of 1 MB = 1,000,000 bytes, 2^{25} bytes = 33.6 MB. Sources that define 1 MB = 2^{20} bytes will arrive at an SDRAM capacity of 32 MB.

[2] The on-chip capacity of the Hammerhead is 2^{19}, or 524,288, bytes. Because we define KB as 1,000 bytes, the on-chip memory capacity of the Hammerhead is 524 KB.

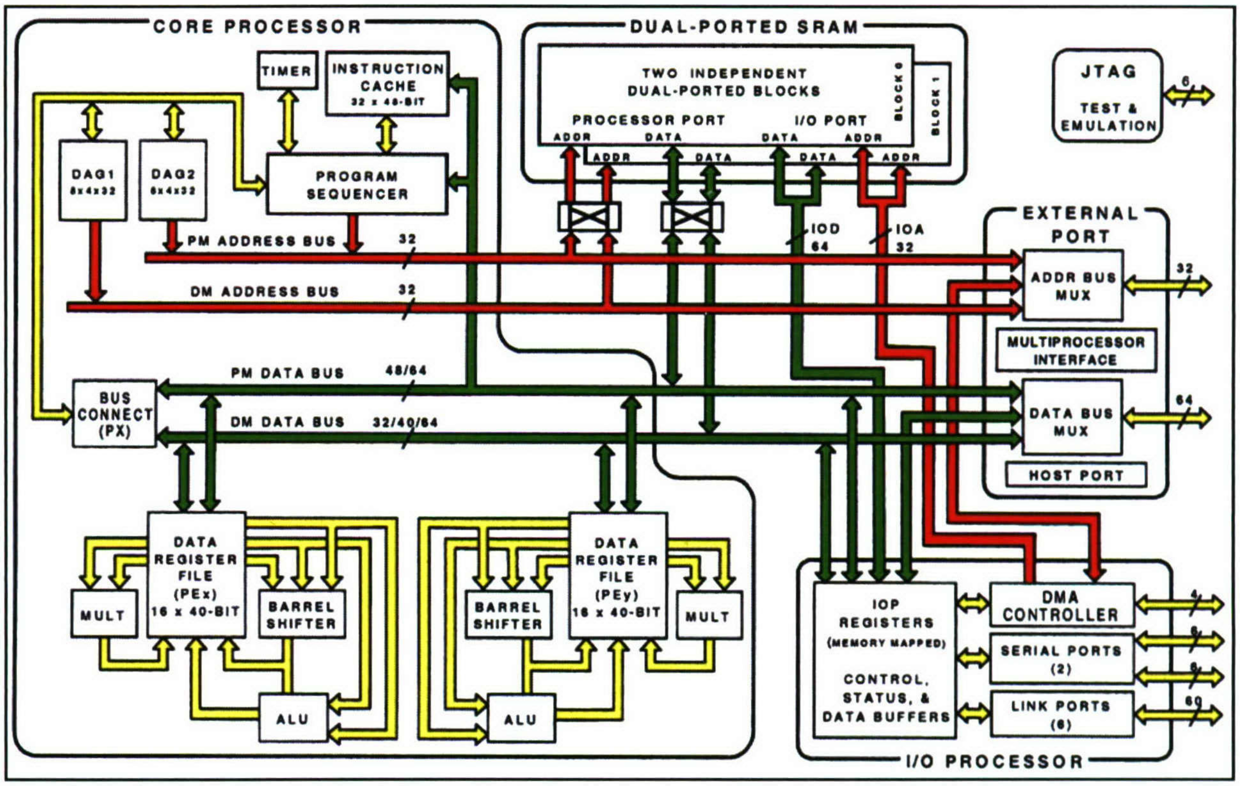

Figure 4: ADSP-21160 Hammerhead internal architecture

These components are connected via two independent address buses and two independent data buses.

2.1.1 Core Processor

The key feature of the core processor is the presence of two ALUs (arithmetic and logic units). These two ALUs must operate in SIMD (single instruction stream, multiple data streams) fashion: in any given clock cycle, the only instruction the second ALU is allowed to execute is the instruction the first ALU is executing.

All instructions in the Hammerhead instruction set can be executed in a single clock cycle. Among them is an instruction that simultaneously computes the sum, difference, and product of two floating-point operands, for a total of three floating-point operations in one clock cycle. This ability is very useful when performing FFTs (fast Fourier transforms), where the basic FFT butterfly requires these three operations. Combining the two ALUs with the ability to perform three floating-point operations in a single cycle gives us the peak throughput of a single 80 MHz Hammerhead: 480 Mflop/s.

2.1.2 On-Chip SRAM

The Hammerhead has 524 KB of on-chip SRAM, arranged in two banks. This SRAM is dual-ported, making it accessible by both the core processor and the I/O processor simultaneously. Furthermore, this on-chip memory is truly random access, in that any location may be accessed with no wait states. This ability is useful during local corner turn operations, when on-chip data must be accessed in strided fashion.

For best performance, one bank of SRAM should be reserved for code, with the data residing in the other bank. This arrangement allows the Hammerhead to retrieve code and data from SRAM simultaneously. Furthermore, code currently being executed on the Hammerhead and data currently being used should be resident in the on-chip SRAM: the performance is heavily penalized if code and data must be retrieved from off-chip DRAM (dynamic random access memory).

2.1.3 I/O Processor

The I/O processor provides background DMA (direct memory access) capabilities to the Hammerhead. It can support the following transfer types:

- between the on-chip SRAM and external memory or external peripherals
- between the on-chip SRAM and the internal memory of other DSPs
- between the on-chip SRAM and a host processor
- between the on-chip SRAM and the serial ports
- between the on-chip SRAM and the link ports
- between external memory and external peripherals

Once it has received a DMA command from the core processor, it can operate completely in the background without additional core processor support. This feature is useful in hiding memory accesses behind computations.

The six link ports, which are part of the I/O processor, are 8 bits wide and were originally designed to be clocked at the processor clock rate. However, currently available Hammerheads have link ports clocked at half the processor clock rate, or 40 MHz.

2.1.4 External Port

The external port is used primarily to connect to external DRAM. The external port's address bus is 32 bits wide, while the data bus is 64 bits wide. In addition to this interface, the external port also has a multiprocessor interface and a host port.

2.2 QUAD-HAMMERHEAD CLUSTER

The Hammerheads in the DR are grouped into three clusters of four Hammerheads (see Figure 5 for a diagram of a single cluster).

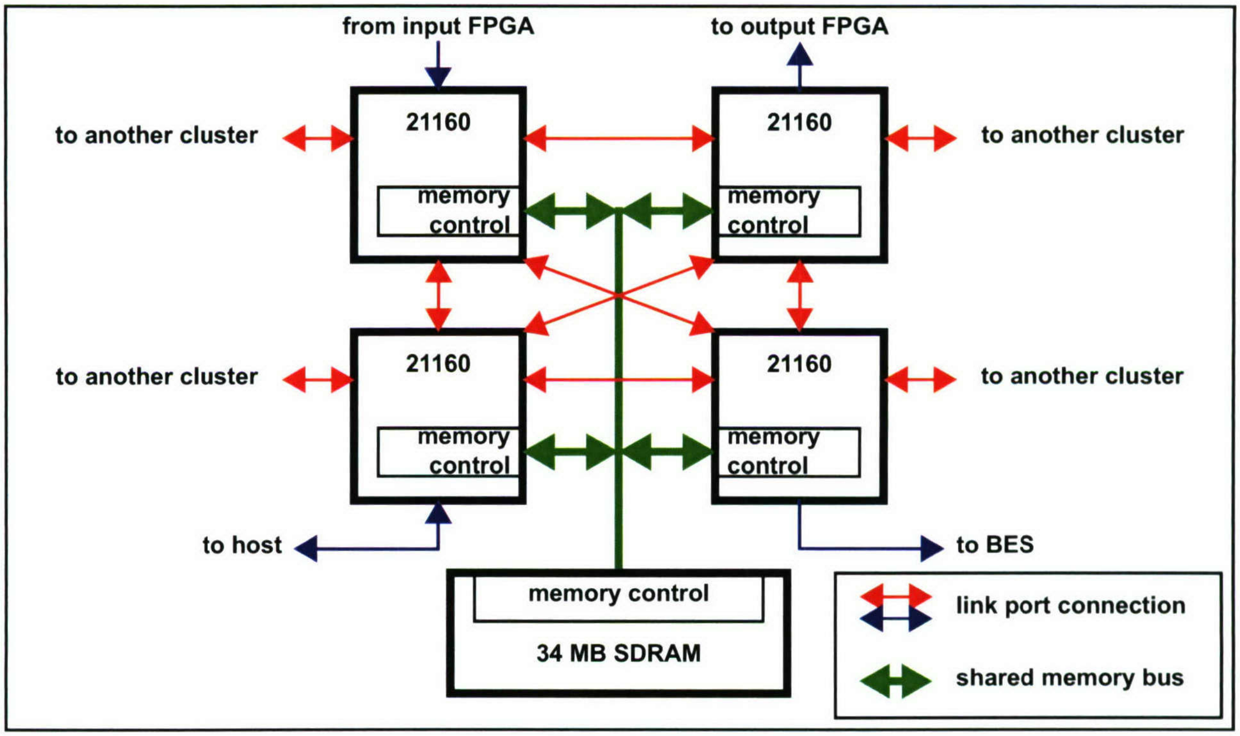

Figure 5: Quad-Hammerhead cluster

As indicated before, each Hammerhead cluster has connections to the input FPGA, the output FPGA, the host, and the BES.

Each cluster has 34 MB of SDRAM, which is accessed via the shared memory bus.

2.3 SHARED MEMORY BUS

The four Hammerheads in a cluster share a memory bus connecting to the common external memory. The memory bus is 64 bits wide, and is clocked at half the Hammerhead clock rate. With a nominal 40 MHz clock rate, the shared bus's peak bandwidth is 320 MB/s.

In practice, the sustained bandwidths are lower. For reads from DRAM, there are three cycles of setup for every four cycles of data access, or four 64-bit words per seven cycles. Therefore, the best sustained bandwidth for reads from DRAM is four-sevenths of 320 MB/s, or 176 MB/s. For writes to DRAM, there is one cycle of setup for every four cycles of data access, or four 64-bit words per five cycles. Therefore, the best sustained bandwidth for writes to DRAM is four-fifths of 320 MB/s, or 256 MB/s. Benchmark results from Bittware show that the actual sustained bandwidth is closer to 240 MB/s.

3. FREQUENCY-DOMAIN DR ALGORITHM DESCRIPTION

The BSAR signal processing algorithm performs the following functions:

- FIR (finite impulse response) filtering
- beamforming
- subband selection
- alias correction
- phase correction

FIR filtering is performed in the frequency domain, and employs the overlap and save filtering method to avoid the computational burden of one long FFT for all the samples and to allow the data to be processed before they are all available. Beamforming and subband selection are also performed in the frequency domain, and the beamforming coefficients are combined with the FIR filter coefficients for computational expediency.

Alias correction is performed when the CBASS subband selection (frequency filter) response is greater than the detection bandwidth. When the BSAR is emulating ADCAP (ADvanced CAPability) processing, FIR filtering followed by decimation in the time domain is equivalent to applying the equivalent filter in the frequency domain (i.e. fast convolution) followed by a frequency folding operation.

Phase correction is necessary to compensate for the overlap and save filtering process and the basebanding oscillator signal.

3.1 ORIGINAL ALGORITHM MODE

3.1.1 Overlap and Save Filtering FFT

The input data are converted into the frequency domain for filter application. Instead of performing one long FFT on the entire sample extent, which would require a large number of operations and a large quantity of memory as well as waiting for all the data to be available before starting, the DR performs shorter FFTs on subsets of the data [1]. Nominally, for the original algorithm mode, 6,432 new samples and 1,760 overlap samples from the previous data block (or zeros if we are processing the first block of data) are transformed using an 8K FFT (see Figure 6). Because the input data are real, a real FFT is used.

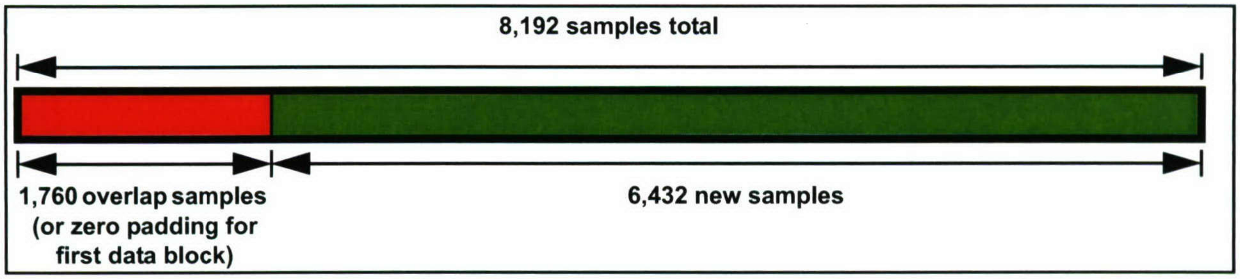
8,192 samples total
1,760 overlap samples (or zero padding for first data block)
6,432 new samples

Figure 6: Overlap and save filtering FFT: original algorithm mode

3.1.2 Subband Selection and Combined Filter Application/Beamforming

Once the data are transformed into the frequency domain, frequency bins corresponding to the subbands of interest are retained. Nominally, for the original algorithm mode, 278 frequency bins from four subbands are retained.

Within each subband, nominally 19 beams are formed for the original algorithm mode. The beam-space output $b_{i,j}$ for beam i at frequency bin j is computed using the following equation:

$$b_{i,j} = \sum_{k=1}^{N_{ch}} c_{j,k} \times w_{i,j,k} \qquad (1)$$

where the number of channels N_{ch} is nominally 52, $c_{j,k}$ is the element-space data for frequency bin j and channel k, and $w_{i,j,k}$ is the beamforming coefficient for beam i, frequency bin j, and channel k.

To reduce the amount of data to be transferred into the DR and the amount of computation to be performed during the real-time receive cycle, the frequency-domain filtering coefficients are combined with the beamforming weights prior to their application to the element-space data.

3.1.3 Alias Correction

If the bandwidth of the filter exceeds the detection bandwidth, the band edges must be aliased in the frequency domain so the resultant bandwidth equals the detection bandwidth. This processing is equivalent to the time domain implementation used in the ADCAP front end processor (FIR filtering followed by downsampling).

Let N_{bins} be the number of frequency bins per subband, and let N_{IFFT} be the length of the IFFT (inverse FFT) to be performed in the next stage, which is equal to the detection bandwidth. The number of frequency bins on each end to be folded is equal to

$$N_{fold} = \frac{N_{bins} - N_{IFFT}}{2} \tag{2}$$

Let F be the pre-aliasing sequence, with N_{bins} frequency bins, and let G be the post-aliasing sequence, with N_{IFFT} frequency bins. Then

$$G(1{:}N_{fold}) = F(N_{fold} + 1{:}2N_{fold}) + F(N_{bins} - N_{fold} + 1{:}N_{bins}) \tag{3}$$

$$G(N_{fold} + 1{:}N_{bins} - N_{fold}) = F(2N_{fold} + 1{:}N_{bins} - 2N_{fold}) \tag{4}$$

$$G(N_{bins} - N_{fold} + 1{:}N_{bins}) = F(1{:}N_{fold}) + F(N_{bins} - 2N_{fold} + 1{:}N_{bins} - N_{fold}) \tag{5}$$

(see Figure 7 below).

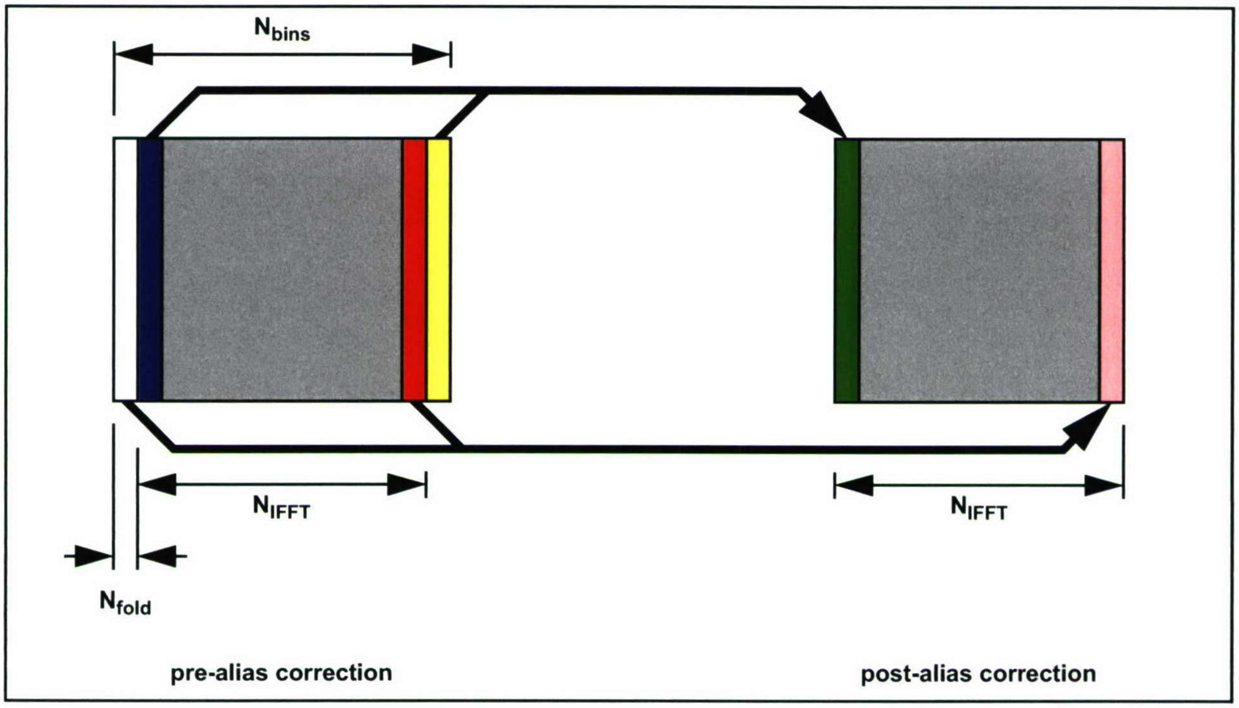

Figure 7: Alias correction

In the original algorithm mode, N_{IFFT} is nominally 256, while N_{bins} is nominally 278.

3.1.4 Overlap and Save Filtering IFFT

After alias correction, the frequency bins in each subband are transformed into the time domain via an IFFT. After the IFFT, those samples corresponding to the overlapping data appended during the FFT stage are removed from the front end of the vector.

If the number of overlap samples appended during the FFT stage is equal to $N_{overlap}$, then the number of samples to be discarded after the IFFT stage is equal to

$$N_{discard} = N_{overlap} \times \frac{N_{IFFT}}{N_{FFT}} \tag{6}$$

where N_{FFT} is the length of the FFT in the overlap and save filtering FFT stage. Nominal values for these parameters for the original algorithm mode are:

- $N_{discard}$: 55
- $N_{overlap}$: 1760
- N_{FFT}: 8,192
- N_{IFFT}: 256

3.1.5 Phase Correction

Phase correction is needed to account for the band selection process in each data frame. Each sample is multiplied by a complex scalar to perform phase correction.

3.2 DEFAULT ALGORITHM

3.2.1 Overlap and Save Filtering FFT

In the new, default algorithm mode, 2,048 new samples and 2,048 overlap samples from the previous data block are transformed using a 4K FFT (see Figure 8).

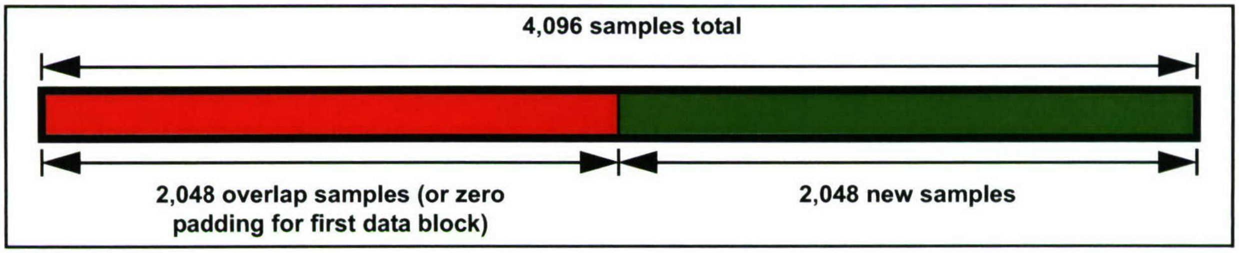
4,096 samples total
2,048 overlap samples (or zero padding for first data block)
2,048 new samples

Figure 8: Overlap and save filtering FFT: default algorithm mode

3.2.2 Subband Selection and Combined Filter Application/Beamforming

Once the data are transformed into the frequency domain, frequency bins corresponding to the subbands of interest are retained. In the default algorithm mode, nominally 140 frequency bins from one subband are retained. Within this single subband, 14 (but possibly as many as 35) beams are formed.

As was done in the original algorithm mode, the frequency-domain filtering coefficients are combined with the beamforming weights prior to their application to the element-space data to reduce the amount of data to be transferred into the DR and the amount of computation to be performed during the real-time receive cycle.

3.2.3 Alias Correction

In the default algorithm mode, alias correction is performed in the same fashion as it is performed in the original algorithm mode. The only differences are in the parameters: in this mode, N_{IFFT} is nominally 128, and N_{bins} is nominally 140.

3.2.4 Overlap and Save Filtering IFFT

The relevant nominal parameters for this stage in this mode are:

- $N_{discard}$: 64
- $N_{overlap}$: 2,048
- N_{FFT}: 4,096
- N_{IFFT}: 128

3.2.5 Phase Correction

Phase correction in this mode is performed identically to phase correction in the original algorithm mode.

4. COMPUTATIONAL WORKLOAD

In this chapter, we derive expressions for the workload for the BSAR signal processing algorithm, and give the computational workload for the nominal parameters for both algorithm modes.

As indicated before, instead of waiting for all of the input samples associated with a single listen interval to be collected, the data are divided into blocks that are processed as they become available to the DR. Therefore, the workload described below is performed for each data block.

4.1 OVERLAP AND SAVE FILTERING FFT

The overlap and save filtering stage uses an N_{FFT}-point FFT. With $N_{overlap}$ samples of overlap between two consecutive data blocks, the number of samples processed per data block is

$$N_{data\ block\ size} = N_{FFT} - N_{overlap} \tag{7}$$

Each N_{FFT}-point real FFT requires n_{FFT} flop, where

$$n_{FFT} = 2.5 N_{FFT} \log_2 N_{FFT} \text{ flop } [2] \tag{8}$$

Given that an N_{FFT}-point FFT must be performed for each of the $N_{channels}$ channels, the total flop count for the FFT portion of overlap and save filtering is

$$\begin{aligned} n_{overlap\ \&\ save\ FFT} &= n_{FFT} \times N_{channels} \text{ flop} \\ &= 2.5 N_{FFT} \log_2 N_{FFT} \times N_{channels} \text{ flop} \end{aligned} \tag{9}$$

After the FFT, only the N_{bins} frequency bins of interest for each of $N_{subbands}$ frequency subband are retained for additional processing.

4.2 COMBINED FILTER APPLICATION/BEAMFORMING

Forming the filtered and beamformed output $A(i)$ for a single beam and a single frequency bin i requires the evaluation of an $N_{channels}$-point dot product between the combined subband filtering and beamforming coefficients β and the subbanded frequency-domain input data S. This single dot product will require

$$n_{dot\ product} = 8 N_{channels} \text{ flop} \tag{10}$$

Computing the filtered and beamformed output for all N_{beams} beams, all $N_{subbands}$ subbands, and all N_{bins} frequency bins will require

$$\begin{aligned} n_{beamform} &= n_{dot\,product} \times N_{beams} \times N_{subbands} \times N_{bins} \text{ flop} \\ &= 8N_{channels} \times N_{beams} \times N_{subbands} \times N_{bins} \text{ flop} \end{aligned} \tag{11}$$

4.3 ALIAS CORRECTION

Alias correction is required only if the number of frequency bins in the subband N_{bins} exceeds the number of points in the IFFT N_{IFFT}. The number of frequency bins to be folded from the beginning of the subband to the end of the subband is equal to

$$N_{fold} = \frac{N_{bins} - N_{IFFT}}{2} \tag{12}$$

The number of frequency bins to be folded from the end of the subband to the beginning of the subband is also equal to N_{fold}.

Folding a single frequency bin will require 2 flop. The number of flop necessary to fold all of the $2N_{fold}$ aliased frequency bins and all $N_{beams} \times N_{subbands}$ beam bands is

$$\begin{aligned} n_{alias} &= 4N_{fold} \times N_{beams} \times N_{subbands} \text{ flop} \\ &= 2 \times (N_{bins} - N_{IFFT}) \times N_{beams} \times N_{subbands} \text{ flop} \end{aligned} \tag{13}$$

4.4 OVERLAP AND SAVE FILTERING IFFT

Performing an N_{IFFT}-point complex IFFT requires

$$n_{IFFT} = 5N_{IFFT}\log_2 N_{IFFT} \text{ flop} \tag{14}$$

Given that an N_{IFFT}-point IFFT must be performed for each of the $N_{beams} \times N_{subbands}$ beam bands, the total flop count for the IFFT portion of overlap and save filtering is equal to

$$\begin{aligned} n_{overlap\ \&\ save\ IFFT} &= n_{IFFT} \times N_{beams} \times N_{subbands} \text{ flop} \\ &= 5N_{IFFT}\log_2 N_{IFFT} \times N_{beams} \times N_{subbands} \text{ flop} \end{aligned} \tag{15}$$

After the IFFT, the first $N_{overlap} \times (N_{IFFT}/N_{FFT})$ samples are discarded as part of the overlap and save filtering process. The remaining $N_{post\,dec}$ samples represent the post-decimation filtered and beamformed time-domain data, where

$$N_{post\,dec} = (N_{FFT} - N_{overlap}) \times \frac{N_{IFFT}}{N_{FFT}} \tag{16}$$

4.5 PHASE CORRECTION

Phase correction entails multiplying each sample in the filtered and beamformed output by the complex phase correction factor $e^{j(m-1)\gamma}$. Both m and γ can be computed outside the real-time loop, so phase correction requires 6 flop for a complex multiply for each sample in the output. The total number of flop needed to correct the phase for all $N_{post\,dec}$ samples and all $N_{beams} \times N_{subbands}$ beam bands is

$$n_{phase\ correction} = 6N_{post\,dec} \times N_{beams} \times N_{subbands}\ \text{flop} \tag{17}$$

These computations may be subsumed into the previous IFFT stage by combining the IFFT weights with the phase correction term.

4.6 COMPUTATIONAL WORKLOAD ANALYSIS

4.6.1 Original Algorithm Mode

The nominal parameters for the original algorithm mode are given below in Table 1.

Table 1: Original Algorithm Mode Nominal Parameters

Parameter	Variable Name	Nominal Value
number of channels	$N_{channels}$	52
FFT length (samples)	N_{FFT}	8,192
FFT overlap length (samples)	$N_{overlap}$	1,760
number of frequency bins in a subband	N_{bins}	278
IFFT length (samples)	N_{IFFT}	256
number of beams	N_{beams}	19
number of subbands	$N_{subbands}$	4

The derived nominal parameters for the original algorithm mode are given below in Table 2.

Table 2: Original Algorithm Mode Derived Nominal Parameters

Parameter	Variable Name	Nominal Value
data block size (samples)	$N_{data\ block\ size}$	6,432
number of folded samples	N_{fold}	11
number of post-decimation samples	$N_{post\ dec}$	201

The flop count for the original algorithm mode is given below in Table 3.

Table 3: Original Algorithm Mode Flop Count

Stage	Variable Name	Flop Count	Fraction of Total
overlap and save FFT	$n_{overlap\ \&\ save\ FFT}$	13,844,480	58.9%
filter and beamform	$n_{beamform}$	8,789,248	37.4%
alias correction	n_{alias}	3,344	0.0%
overlap and save IFFT	$n_{overlap\ \&\ save\ IFFT}$	778,240	3.3%
phase correction	$n_{phase\ correction}$	91,656	0.4%
Total		**23,506,968**	**100.0%**

4.6.2 Default Algorithm Mode

The nominal parameters for the default algorithm mode are given below in Table 4.

Table 4: Default Algorithm Mode Nominal Parameters

Parameter	Variable Name	Nominal Value
number of channels	$N_{channels}$	52
FFT length (samples)	N_{FFT}	4,096
FFT overlap length (samples)	$N_{overlap}$	2,048
number of frequency bins in a subband	N_{bins}	140
IFFT length (samples)	N_{IFFT}	128
number of beams	N_{beams}	14
number of subbands	$N_{subbands}$	1

The derived nominal parameters for the default algorithm mode are given below in Table 5.

Table 5: Default Algorithm Mode Derived Nominal Parameters

Parameter	Variable Name	Nominal Value
data block size (samples)	$N_{data\ block\ size}$	2,048
number of folded samples	N_{fold}	7
number of post-decimation samples	$N_{post\ dec}$	128

The flop count for the default algorithm mode is given below in Table 6.

Table 6: Default Algorithm Mode Flop Count

Stage	Variable Name	Flop Count	Fraction of Total
overlap and save FFT	$n_{overlap\ \&\ save\ FFT}$	6,389,760	87.8%
filter and beamform	$n_{beamform}$	815,360	11.2%
alias correction	n_{alias}	336	0.0%
overlap and save IFFT	$n_{overlap\ \&\ save\ IFFT}$	62,720	0.9%
phase correction	$n_{phase\ correction}$	5,376	0.1%
Total		**7,273,552**	**100.0%**

4.7 THROUGHPUT REQUIREMENT

4.7.1 Original Algorithm Mode

In an active sonar time line, the torpedo transmits a sonar waveform (nominally 264 ms in duration), then receives sonar data for several seconds before beginning the next sonar cycle. Instead of collecting input data for several seconds, then processing them all at once, the data are divided into data blocks and processed as they arrive at the DR (see Figure 9). The combined beamforming and filtering coefficients may be computed during the transmit time, before the end of which input data cannot be processed.

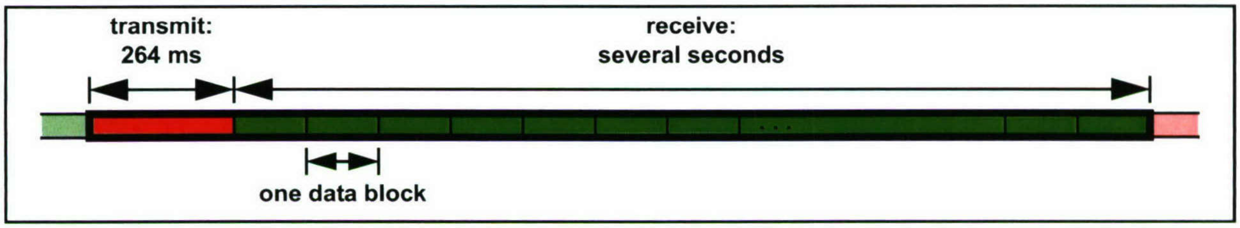

Figure 9: Time line for an active sonar cycle

To compute the sustained computational throughput requirement, the total time available to perform the BSAR computational workload is needed. We shall use as the time available the amount of time necessary to collect $N_{data\ block\ size}$ samples at the nominal sampling frequency of $f_{sample} = 100$ kHz :

$$t_{sample} = \frac{N_{data\ block\ size}}{f_{sample}} \tag{18}$$

With the nominal parameters for the original algorithm mode, t_{sample} is 64.3 ms.

The CBASS program places a 50% spare processor throughput requirement on the BSAR to ensure sufficient margins and to avoid cost and schedule overruns: the amount of spare throughput must be at least 50% of the throughput consumed. Given that the 64.3 ms total time for the original algorithm mode needs to be 150% of the time available for computation, the real-time execution time is 42.9 ms. Based on this reduced available time, the sustained throughput requirement for the original algorithm mode is 548.2 Mflop/s.

As long as the number of data samples processed in a single data block $N_{data\ block\ size}$ remains constant, the throughput requirements will not change: shorter or longer receive intervals will not affect our sustained throughput requirement. Of course, the sustained throughput requirement is proportional to the sampling frequency f_{sample}.

4.7.2 Default Algorithm Mode

The nominal parameters for the default algorithm mode result in a t_{sample} of 20.5 ms. Because this time is 150% of the time available for computation, the real-time execution time is 13.7 ms. Based on this reduced available time, the sustained throughput requirement for the default algorithm mode is 532.7 Mflop/s.

5. PROCESSOR ASSESSMENT METHODOLOGY

5.1 MOTIVATION

Based on the measured sustained throughput of the Hammerhead on the types of computations performed in the BSAR algorithm, and the amount of computational work the algorithm entails, we may compute an estimated execution time.

In the original algorithm mode, the overlap and save FFT stage requires 13.8 Mflop. The peak throughput of an 80 MHz Hammerhead on FFTs is 480 Mflop/s. With 12 Hammerheads total and a measured efficiency of 65% on real 8K FFTs, we estimate the sustained throughput of the DR to be 3.74 Gflop/s. Given these numbers, we would expect the overlap and save FFT stage to take

$$13{,}844{,}480 \text{ flop} \div \frac{3.74 \times 10^{9} \text{ flop}}{\text{second}} = 3.70 \times 10^{-3} \text{ s} \quad (19)$$

We define the efficiency metric as the ratio of the sustained throughput, or the computational performance we actually achieved, to the peak throughput, or the theoretical best computational performance:

$$\text{efficiency} = \frac{\text{sustained throughput}}{\text{peak throughput}} \quad (20)$$

The beamforming/filtering stage in the original algorithm mode requires 8.8 Mflop. The peak throughput of an 80 MHz Hammerhead on matrix computations is 320 Mflop/s. With 12 Hammerheads total and a measured efficiency of 38%, we estimate the sustained throughput of the DR to be 1.46 Gflop/s. Given these numbers, we would expect the beamforming stage to take

$$8{,}789{,}248 \text{ flop} \div \frac{1.46 \times 10^{9} \text{ flop}}{\text{second}} = 6.02 \times 10^{-3} \text{ s} \quad (21)$$

We can use this process to compute the total estimated execution time for the original algorithm mode in this manner (see Table 7):

Table 7: Naïve Estimated Execution Time for the Original Algorithm Mode

Stage	Flop Count	DR Sustained Throughput (Gflop/s)	Estimated Execution Time (ms)
overlap and save FFT	13,844,480	3.74	3.70
filter and beamform	8,789,248	1.46	6.02
alias correction	3,344	1.46	0.00
overlap and save IFFT	778,240	3.51	0.22
phase correction	91,656	1.46	0.06
Total	**23,506,968**	-	**10.01**

This estimated execution time - 10.01 ms - is only a small fraction of the 42.9 ms we have available to us. If we naïvely consider only the computational throughput requirements, then we might come to the conclusion that the real-time deadlines can be easily met. Our methodology represents a more sophisticated analysis that also considers the impact of communication and contention on the execution time.

5.2 PROCESSOR ASSESSMENT METHODOLOGY

In our methodology, we propose several mappings of the BSAR algorithm onto the DR architecture, then estimate the performance of the DR using those mappings (see Figure 10). The estimates of the execution time consider not only the time necessary to perform the computations but also the time needed to move the requisite data into memory and to communicate data between processors. Based on a preliminary execution time using estimated computation and communication efficiencies, we winnow down the field of candidate mappings to the most promising ones. For those remaining mappings, the methodology calls for a refinement of our estimate of the execution time by replacing estimates of the efficiencies with efficiencies measured using benchmarks. We also perform sensitivity analyses to gauge the degree to which our execution time estimates vary if we vary our efficiency estimates.

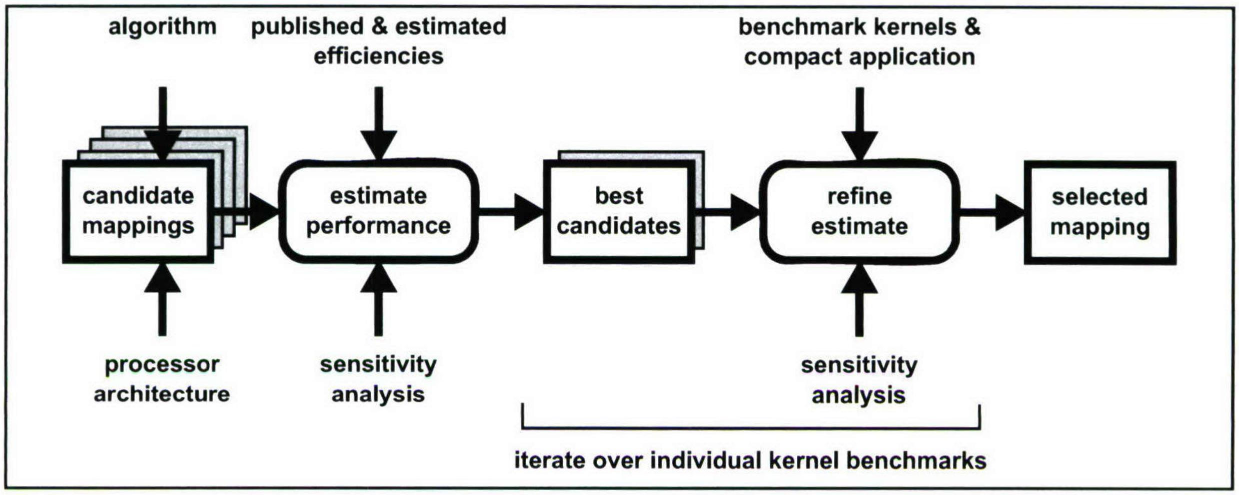

Figure 10: Processor assessment methodology

In the original version of this report, we considered three candidate mappings and estimated their performance based on published and estimated efficiencies. We also reported the results of a sensitivity analysis performed on the mapping that resulted in the best estimated performance. In this revision, we focus our analysis to the mapping that resulted in the best estimated performance and has been adopted by the BSAR design teams.

6. PROPOSED MAPPING AND ESTIMATED TIMING

Because the processing on each quad-Hammerhead cluster is nearly identical, we will concentrate our mapping discussions to a single quad-Hammerhead cluster.

The efficiencies given in this chapter are based on benchmark experiments performed on a Bittware quad-Hammerhead board whose architecture closely resembles that of a DR quad-Hammerhead cluster. Given these measured efficiencies, the proposed mapping results in an predicted execution time of 39.8 ms for the original algorithm mode, which is within the 42.9 ms available. For the default algorithm mode, the predicted execution time is 6.4 ms, which is within the 13.7 ms available.

6.1 ORIGINAL ALGORITHM MODE

6.1.1 Data Input

The 52 channels of data are distributed over the three clusters, with one cluster receiving 18 channels of data and the other two clusters receiving 17 channels of data each. As the input data are real and not complex, each sample occupies four bytes. With 6,432 samples per data block, the quantity of data that must be brought into a cluster in the worst case is 463 KB.

The input FPGA needs no appreciable amount of memory, as the input data effectively trickle in at the nominal sampling rate of 100 kHz. Because constant accesses to the shared DRAM to save the data one sample at a time would prevent efficient accesses to DRAM by the other Hammerheads, we will dedicate one Hammerhead - the Hammerhead with the link port connected to the input FPGA - to data input. This Hammerhead will accumulate several input samples in SRAM and copy them into DRAM in a single transfer.

To make the most efficient use of the Hammerhead's on-chip SRAM, half of the 524 KB will be reserved for instructions, while the other 262 KB will be made available for data storage. To allow the input Hammerhead to accept data from the input FPGA while simultaneously writing data to DRAM, we will establish two buffers, each capable of storing $1/16\text{th}$ of the sample extent, or 402 samples, for all 18 channels. Each buffer will therefore be 29 KB in size.

The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy $1/16\text{th}$ of the data into DRAM is

$$28{,}944 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 121 \times 10^{-6} \text{ s} \tag{22}$$

One such copy to DRAM will occur every 402 samples, or 4.02×10^{-3} s. The total time spent copying data into DRAM is $1{,}930 \times 10^{-6}$ s per data block.

6.1.2 Overlap and Save Filtering FFT

For each channel's worth of data, an 8K-point real FFT must be performed as the first step in the overlap and save filtering. As one of the four Hammerheads in each cluster is dedicated to data input, the signal processing computations will be performed by the remaining three Hammerheads. Each Hammerhead's share of the data for this stage, in the worst case, is six channels.

On-Chip Memory Usage. The 6,432 samples of new data plus the 1,760 samples of old overlap data will occupy 33 KB. Even though only 33 KB are copied from DRAM to SRAM for a single vector, the input data vector will occupy twice that space - 66 KB - because the vector must be large enough to store the complex results of the in-place FFT.

To keep the processor busy performing FFTs, we will set up two separate buffers, each 66 KB in size. While the processor is performing an FFT on the data in one buffer, the other buffer will either be filled with more data from DRAM or copied back into DRAM. These two buffers together will occupy 131 KB of the on-chip SRAM. We will also need to store in SRAM the FFT weights, which will occupy 33 KB.

Memory Access: DRAM to SRAM. The 33 KB of input data must be copied from DRAM to SRAM over the shared memory bus. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the input data from DRAM is

$$32{,}768 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 186 \times 10^{-6} \text{ s} \tag{23}$$

FFT Processing. A single 8K-point real FFT will require 266,240 flop. The measured sustained throughput of the Hammerhead on a real 8K FFT is 312 Mflop/s. At this rate, the time for a single 8K-point real FFT is

$$266{,}240 \text{ flop} \div \frac{312 \ \times 10^{6} \text{ flop}}{\text{second}} = 853 \times 10^{-6} \text{ s} \tag{24}$$

Memory Access: SRAM to DRAM. After the 8K-point FFT, we are only interested in 278 frequency bins for each of four subbands, which together will occupy 9 KB. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the frequency-domain data to DRAM is

$$8{,}896 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 37 \times 10^{-6} \text{ s} \tag{25}$$

Simultaneous Operation of Three Hammerheads in a Cluster. The three Hammerheads performing signal processing in a single Hammerhead cluster may perform FFTs simultaneously. However, because they share the cluster memory bus, they must access DRAM sequentially. Therefore, the second Hammerhead cannot begin loading its data from DRAM into its first buffer in SRAM until the first Hammerhead has finished loading its data (see Figure 11).

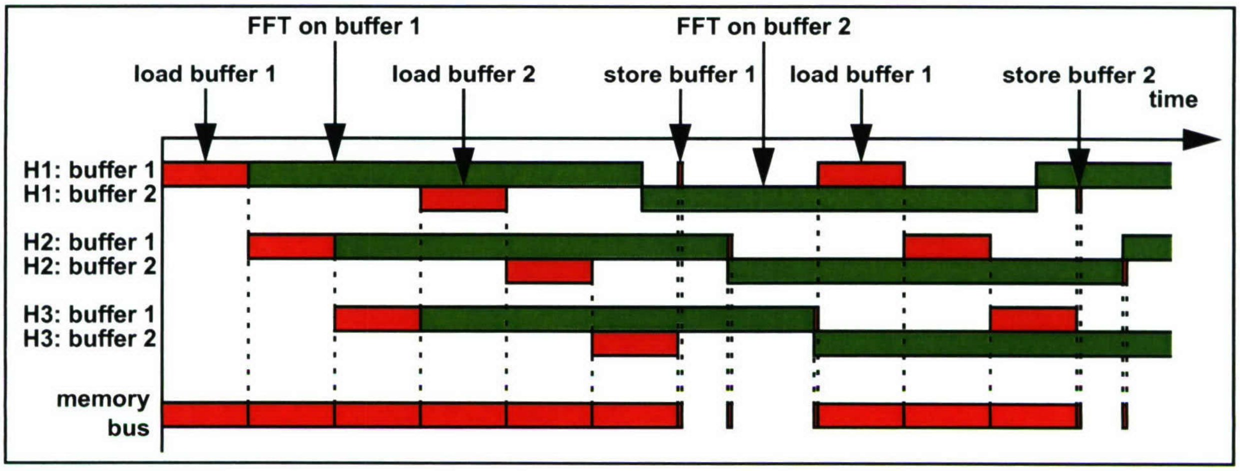

Figure 11: Sequential access of DRAM during overlap and save FFT

Once the third Hammerhead has finished loading its data from DRAM into its first buffer in SRAM, the first Hammerhead may begin loading its second buffer in SRAM. Because the Hammerhead has a dual-ported SRAM and an I/O processor that is independent from the ALU, it can load the second buffer while still processing the first buffer. Furthermore, because the second buffer is expected to be fully loaded before processing on the first buffer is completed, the Hammerhead can begin processing the second buffer as soon as it has completed processing the first buffer. Copying data from a buffer in SRAM to DRAM can be done concurrently with processing as well.

The first Hammerhead cluster processes 18 channels, while the second and third clusters process 17 channels each. In the first Hammerhead cluster, all three Hammerheads process six channels. The last processor to complete its processing will be the third Hammerhead in the first cluster (see Figure 12).

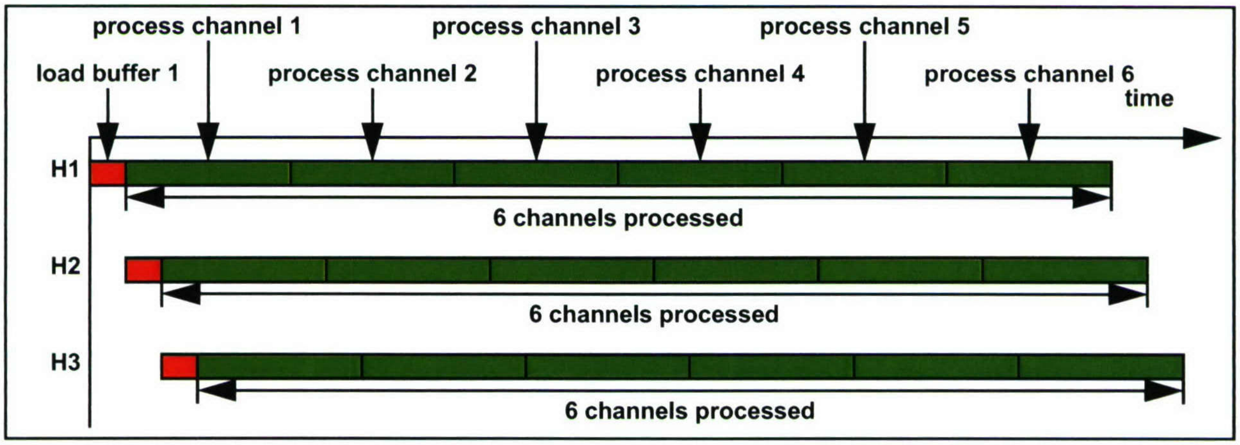

Figure 12: FFT stage processing time line

Aside from the initial copy from DRAM into the first buffer in SRAM and the final copy from SRAM to DRAM, all memory accesses during the FFT stage can be overlapped with computation. We can now estimate the total execution time for this stage, which will be dictated by the third Hammerhead in the first cluster:

- wait for the first Hammerhead to load an 8K-point real vector: 186×10^{-6} s
- wait for the second Hammerhead to load an 8K-point real vector: 186×10^{-6} s
- load an 8K-point real vector: 186×10^{-6} s
- perform 8K-point real FFTs for six channels: $5{,}120 \times 10^{-6}$ s
- store four 278-point complex vectors: 37×10^{-6} s

for a total of $5{,}716 \times 10^{-6}$ s.

6.1.3 Beamforming and Filtering: Step 1

Each Hammerhead forms partial beamforming sums from five or six channels' worth of data, then redistributes the data prior to completing the beamforming operation. Fundamentally, the beamforming operation for a single beam and a single frequency bin involves the following operation:

$$b = \sum_{i=1}^{52} c_i \times w_i \tag{26}$$

where b is the beam-space output, c_i is the element-space data for the ith channel, and w_i is the beamforming weight for the ith channel.

This sum can be divided up by the channels local to each Hammerhead cluster:

$$b = sum_1 + sum_2 + sum_3 \tag{27}$$

$$sum_1 = \sum_{i=1}^{18} c_i \times w_i \tag{28}$$

$$sum_2 = \sum_{i=19}^{35} c_i \times w_i \tag{29}$$

$$sum_3 = \sum_{i=36}^{52} c_i \times w_i \tag{30}$$

Each of sum_1, sum_2, and sum_3 can be further subdivided by channels assigned to each Hammerhead:

$$sum_1 = \sum_{i=1}^{6} c_i \times w_i + \sum_{i=7}^{12} c_i \times w_i + \sum_{i=13}^{18} c_i \times w_i \tag{31}$$

$$sum_2 = \sum_{i=19}^{24} c_i \times w_i + \sum_{i=25}^{30} c_i \times w_i + \sum_{i=31}^{35} c_i \times w_i \tag{32}$$

$$sum_3 = \sum_{i=36}^{41} c_i \times w_i + \sum_{i=42}^{47} c_i \times w_i + \sum_{i=48}^{52} c_i \times w_i \tag{33}$$

To compute each of the nine sums in Equation 31 through Equation 33, we use the following mapping proposed by NUWC (the Naval Undersea Warfare Center) for each Hammerhead:

- Copy the element-space data matrix for the five or six channels local to each processor and all N_{bins} frequency bins from DRAM to SRAM.
- For each of the $N_{beams} \times N_{subbands}$ beam bands:
  - Copy the N_{bins}-point vector with the beamforming/filtering weights for the first channel from DRAM to SRAM.
  - Repeat for all five or six channels: Multiply the weights vector with the corresponding vector from the data matrix and add this product vector to the partial beamforming sum vector while simultaneously copying the weights for the next channel from DRAM to SRAM.
  - Copy the N_{bins}-point vector with the partial sums for these five or six channels from SRAM to DRAM.

On-Chip Memory Usage. The element-space data for six channels and 278 frequency bins will occupy 13 KB. The beamforming/filtering weights vectors for two channels (we will use one weights vector while loading the next weights vector) will occupy 4 KB. The partial beamforming sum vector will require 2 KB. Together, these data products will require 20 KB.

Memory Access: DRAM to SRAM. The element-space data for six channels and 278 frequency bins will occupy 13 KB. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the element-space data from DRAM is

$$13{,}344 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 76 \times 10^{-6} \text{ s} \tag{34}$$

The beamforming/filtering weights for one channel will occupy 2 KB. The time to copy the weights from DRAM is

$$2{,}224 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 13 \times 10^{-6} \text{ s} \tag{35}$$

Beamforming. Adding the partial beamforming sums from one channel for all 278 frequency bins will require 2,224 flop. The measured sustained throughput of the Hammerhead on a multiply-accumulate operation of this size is 121.6 Mflop/s. At this rate, the time for this multiply-accumulate operation is

$$2{,}224 \text{ flop} \div \frac{121.6 \times 10^{6} \text{ bytes}}{\text{second}} = 18 \times 10^{-6} \text{ s} \tag{36}$$

Memory Access: SRAM to DRAM. The partial beamforming sum vector will occupy 2 KB. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the partial sum vector to DRAM is

$$2{,}224 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 9 \times 10^{-6} \text{ s} \tag{37}$$

Simultaneous Operation of Three Hammerheads in a Cluster. The three Hammerheads performing signal processing in a single Hammerhead cluster may perform matrix multiplication/accumulation simultaneously. However, because they share the cluster memory bus, they must access DRAM sequentially. Therefore, the second Hammerhead cannot begin loading its data from DRAM into SRAM until the first Hammerhead has finished loading its data (see Figure 13).

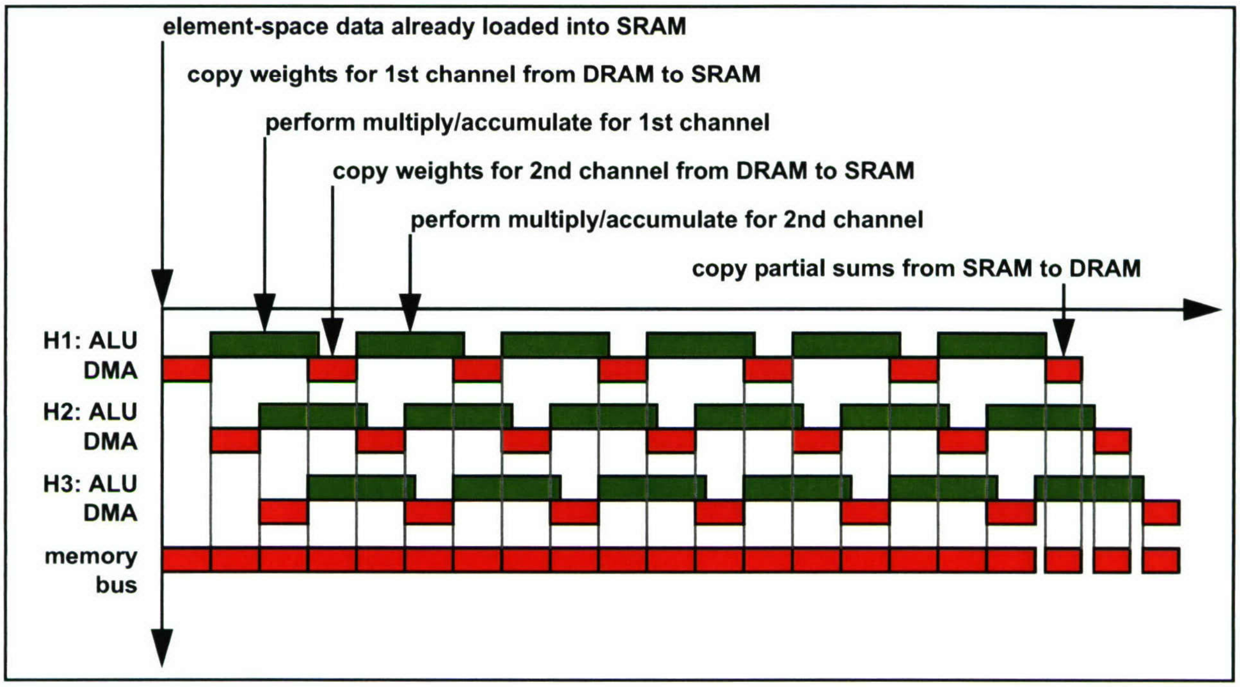
element-space data already loaded into SRAM
copy weights for 1st channel from DRAM to SRAM
perform multiply/accumulate for 1st channel
copy weights for 2nd channel from DRAM to SRAM
perform multiply/accumulate for 2nd channel
copy partial sums from SRAM to DRAM
H1: ALU
DMA
H2: ALU
DMA
H3: ALU
DMA
memory
bus

Figure 13: Sequential access of DRAM during beamforming: step 1

Because the computation time is less than three times (because there are three Hammerheads sharing the memory bus) the time needed to load weights from DRAM, the total time for the first step of beamforming will be almost entirely limited by the shared memory bus bandwidth. The last processor to complete its processing will be the third Hammerhead in each cluster. We can now estimate the total execution time for the first step of beamforming, which will be dictated by the third Hammerhead:

- wait for the first Hammerhead to load the element-space data for six channels: 76×10^{-6} s
- wait for the second Hammerhead to load the element-space data for six channels: 76×10^{-6} s
- load the element-space data for six channels: 76×10^{-6} s
- for each beam band:
  - load the beamforming/filtering weights for all six channels for all three Hammerheads: $13 \times 10^{-6}\text{ s} \times 3 \times 6 = 227 \times 10^{-6}$ s
  - perform multiply/accumulate for the sixth channel: 18×10^{-6} s
  - store partial beamforming sum in DRAM: 9×10^{-6} s

for a total of $19{,}608 \times 10^{-6}$ s for 76 beam bands.

6.1.4 Beamforming/Filter Application Step 2

In the second step of beamforming/filtering, we accumulate the three partial sums that were computed within a single Hammerhead cluster (see Equation 31 through Equation 33). Specifically, each Hammerhead loads from DRAM all three partial sums for one-third of the $N_{beams} \times N_{subbands}$ beam bands, where each input partial sum is computed from six channels' worth of element-space data, and combines them to form one set of output partial sums for one-third of the beam bands.

Before we load the three partial sums computed during the first step of beamforming/filtering, we need to synchronize the three Hammerheads in each cluster to force the processors to wait for the previous step to complete.

On-Chip Memory Usage. Because of the memory limitations of the on-chip SRAM, we will process half of the 278 frequency bins at a time. One-third of 76 beam bands, rounded up, is 26 beam bands. The three sets of input partial sums for 26 beam bands and 139 frequency bins will occupy 87 KB. The output partial sums for 26 beam bands and 139 frequency bins will occupy 29 KB. Together, these data products will require 116 KB.

Memory Access: DRAM to SRAM. The three sets of input partial sums will occupy 87 KB. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the input partial sums from DRAM is

$$\text{86,736 bytes} \div \frac{176 \times 10^{6}\ \text{bytes}}{\text{second}} = 493 \times 10^{-6}\ \text{s} \tag{38}$$

Beamforming. Forming the cluster-wide partial sums for 26 beam bands and 139 frequency bins will require 14,456 flop. The measured sustained throughput of the Hammerhead on matrix operations of this size is 38 Mflop/s. At this rate, the time for the second step of beamforming and filter application is

$$\text{14,456 flop} \div \frac{38 \times 10^{6}\ \text{flop}}{\text{second}} = 376 \times 10^{-6}\ \text{s} \tag{39}$$

Memory Access: SRAM to DRAM. The cluster-wide partial sums for 26 beam bands and 139 frequency bins will occupy 29 KB. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the partial sums to DRAM is

$$\text{28,912 bytes} \div \frac{240 \times 10^{6}\ \text{bytes}}{\text{second}} = 120 \times 10^{-6}\ \text{s} \tag{40}$$

Simultaneous Operation of Three Hammerheads in a Cluster. The three Hammerheads performing signal processing in a single Hammerhead cluster may add the partial sums simultaneously. However, because they share the cluster memory bus, they must access DRAM sequentially. Therefore, the second Hammerhead cannot begin loading its data from DRAM into SRAM until the first Hammerhead has finished loading its data. Furthermore, because the data access time exceeds the computation time, virtually all of the computation time can be hidden behind the data access time (see Figure 14).

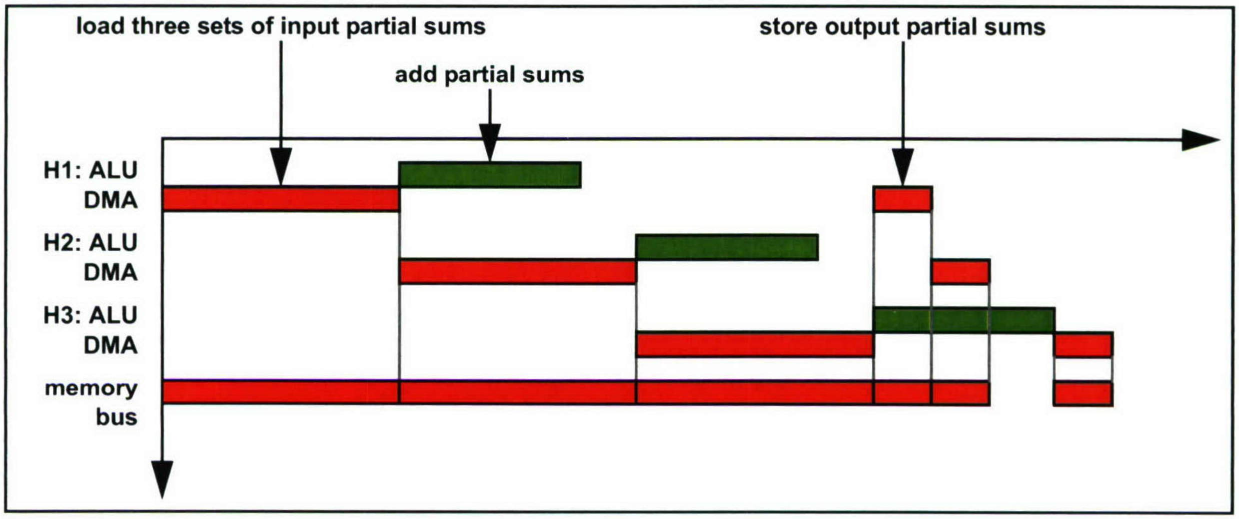
load three sets of input partial sums
store output partial sums
add partial sums
H1: ALU
DMA
H2: ALU
DMA
H3: ALU
DMA
memory
bus

Figure 14: Sequential access of DRAM during beamforming: step 2

We can now estimate the total execution time for the second step of beamforming, which will be dictated by the third Hammerhead:

- wait for the first Hammerhead to load its three sets of input partial sums: 493×10^{-6} s
- wait for the second Hammerhead to load its three sets of input partial sums: 493×10^{-6} s
- load the three sets of input partial sums: 493×10^{-6} s
- add the three partial sums: 376×10^{-6} s
- store the output partial sum in DRAM: 120×10^{-6} s

for a total of $3{,}951 \times 10^{-6}$ s for all 278 frequency bins.

6.1.5 Beamforming and Filtering: Inter-Cluster Communication

Prior to the final step of beamforming and filter application, inter-cluster communication is necessary to exchange beamforming partial sums between clusters. At the end of the beamforming and filter application stage, we would like to have the beams equally distributed over the three Hammerhead clusters, with the entire frequency subband in a single Hammerhead cluster. This arrangement will allow for the remaining stages of the BSAR processing to be done local to a single Hammerhead.

Before we transfer the partial sums between the Hammerhead clusters, we need to synchronize all nine Hammerheads performing signal processing to force the processors to wait for the previous step to complete.

Prior to the inter-cluster communication, each cluster has a partial sum for all 76 beam bands and 278 frequency bins (see Figure 15).

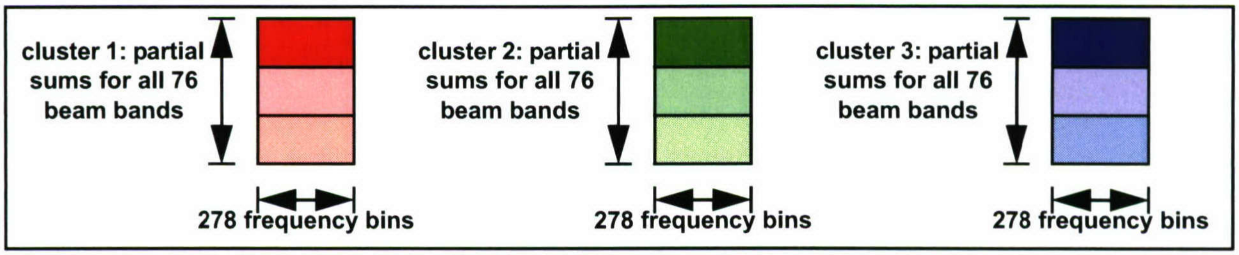
cluster 1: partial sums for all 76 beam bands
cluster 2: partial sums for all 76 beam bands
cluster 3: partial sums for all 76 beam bands
278 frequency bins
278 frequency bins
278 frequency bins

Figure 15: Data distribution prior to the inter-cluster communication

After the inter-cluster communication, each cluster will have all three partial sums for a third of the 76 beam bands (see Figure 16).

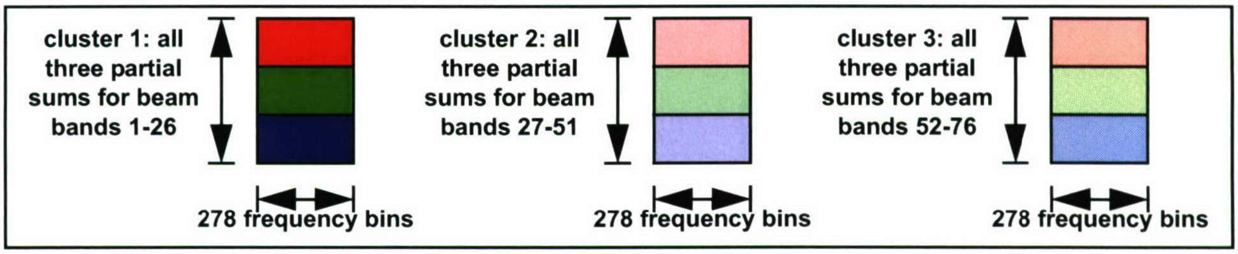
cluster 1: all three partial sums for beam bands 1-26
cluster 2: all three partial sums for beam bands 27-51
cluster 3: all three partial sums for beam bands 52-76
278 frequency bins
278 frequency bins
278 frequency bins

Figure 16: Data distribution after the inter-cluster communication

This communication operation entails sending, in the worst case, 26 beam bands of partial sums to one Hammerhead cluster and 26 beam bands of partial sums to the other Hammerhead cluster. This operation can be performed in two stages: data transfer to the neighboring cluster to the left, and data transfer to the neighboring cluster to the right (see Figure 17).

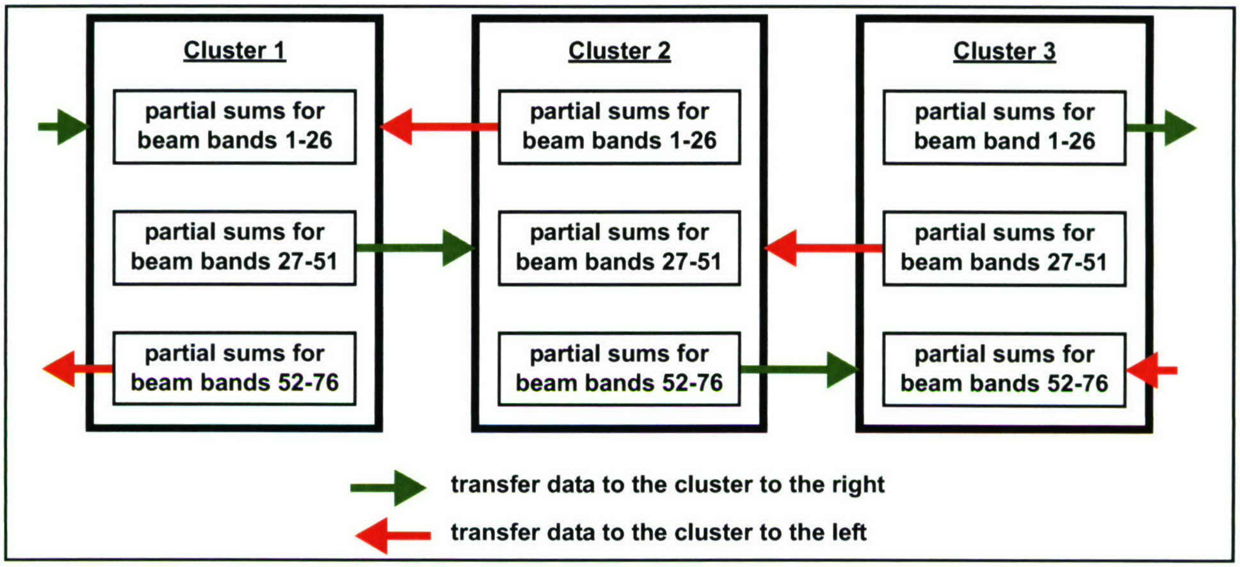
Cluster 1
partial sums for beam bands 1-26
partial sums for beam bands 27-51
partial sums for beam bands 52-76
Cluster 2
partial sums for beam bands 1-26
partial sums for beam bands 27-51
partial sums for beam bands 52-76
Cluster 3
partial sums for beam band 1-26
partial sums for beam bands 27-51
partial sums for beam bands 52-76
transfer data to the cluster to the right
transfer data to the cluster to the left

Figure 17: Inter-cluster communication to exchange beamforming partial sums

In both directions, the worst case quantity of data to be transferred is 58 KB for 26 partial sums and 278 frequency bins.

On-Chip Memory Usage. Because of the memory limitations of the on-chip SRAM, we will transfer half of the 278 frequency bins at a time. The two outgoing sets of partial sums will require 58 KB. The two incoming sets of partial sums will require an additional 58 KB, for a total of 116 KB.

Memory Access: DRAM to SRAM. The two sets of outgoing partial sums occupy 58 KB. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the outgoing data from DRAM is

$$57{,}824 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 329 \times 10^{-6} \text{ s} \tag{41}$$

Inter-Cluster Communication. The measured sustained bandwidth of the link ports is 32 MB/s. At this rate, the time to perform these two transfers is

$$57{,}824 \text{ bytes} \div \frac{32 \times 10^{6} \text{ bytes}}{\text{second}} = 1{,}807 \times 10^{-6} \text{ s} \tag{42}$$

Memory Access: SRAM to DRAM. The two sets of incoming partial sums occupy 58 KB. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy both sets of partial sums to DRAM is

$$57{,}824 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 241 \times 10^{-6} \text{ s} \qquad (43)$$

The total time needed to transfer all 278 frequency bins is

$$2 \times (329 + 1{,}807 + 241) \times 10^{-6} \text{ s} = 4{,}753 \times 10^{-6} \text{ s} \qquad (44)$$

6.1.6 Beamforming/Filter Application Step 3

In the third step of beamforming/filtering, we combine the partial sums across the three Hammerhead clusters. Specifically, each of the nine Hammerheads performing signal processing will combine the three partial sums for one-ninth of the beam bands to produce the total beamformed sum.

Before we load the three partial sums computed during the first two step of beamforming/filtering, we need to synchronize all nine Hammerheads performing signal processing to force the processors to wait for the previous step to complete.

On-Chip Memory Usage. One-ninth of the 76 beam bands, rounded up, is nine beam bands. The three sets of partial sums for nine beam bands and 278 frequency bins will occupy 60 KB. The output total sums for nine beam bands and 278 frequency bins will occupy 20 KB. Together, these data products will require 80 KB.

Memory Access: DRAM to SRAM. The three sets of partial sums will occupy 60 KB. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the partial sums from DRAM is

$$60{,}048 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 341 \times 10^{-6} \text{ s} \qquad (45)$$

Beamforming. Forming the beamformed output for nine beam bands and 278 frequency bins will require 10,008 flop. The measured sustained throughput of the Hammerhead on matrix operations of this size is 38 Mflop/s. At this rate, the time for the third step of beamforming and filter application is

$$10{,}008 \text{ flop} \div \frac{38 \times 10^{6} \text{ flop}}{\text{second}} = 261 \times 10^{-6} \text{ s} \qquad (46)$$

Memory Access: SRAM to DRAM. The beamformed output for 278 frequency bins will occupy 20 KB of memory. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the beamformed output back to DRAM is

$$20{,}016 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 83 \times 10^{-6} \text{ s} \qquad (47)$$

Simultaneous Operation of Three Hammerheads in a Cluster. The three Hammerheads performing signal processing in a single Hammerhead cluster may add the partial sums simultaneously. However, because they share the cluster memory bus, they must access DRAM sequentially. Therefore, the second Hammerhead cannot begin loading its data from DRAM into SRAM until the first Hammerhead has finished loading its data. Furthermore, because the data access time exceeds the computation time, virtually all of the computation time can be hidden behind the data access time (see Figure 18).

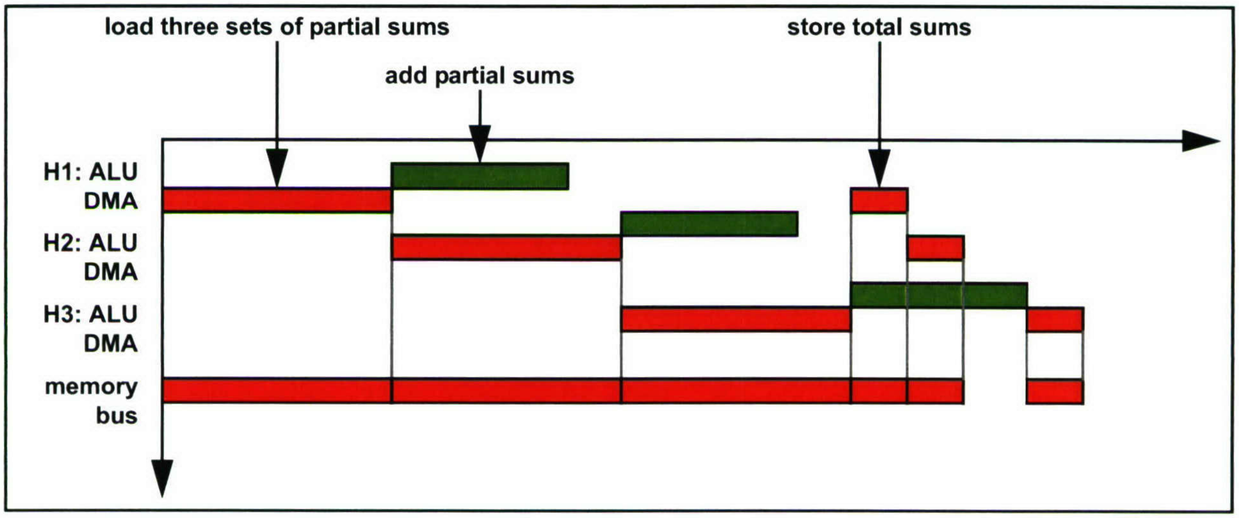

Figure 18: Sequential access of DRAM during beamforming: step 3

We can now estimate the total execution time for the third step of beamforming, which will be dictated by the third Hammerhead:

- wait for the first Hammerhead to load its three sets of partial sums: 341×10^{-6} s
- wait for the second Hammerhead to load its three sets of partial sums: 341×10^{-6} s
- load the three sets of partial sums: 341×10^{-6} s
- add the three partial sums: 261×10^{-6} s
- store the output total sum: 83×10^{-6} s

for a total of $1{,}368 \times 10^{-6}$ s for all 278 frequency bins.

We can now calculate the projected execution time for the entire beamforming/filtering stage:

- step 1: $19{,}608 \times 10^{-6}$ s
- step 2: $3{,}951 \times 10^{-6}$ s
- inter-cluster communication: $4{,}753 \times 10^{-6}$ s
- step 3: $1{,}368 \times 10^{-6}$ s

for a total of $29{,}680 \times 10^{-6}$ s .

6.1.7 Alias Correction

For the alias correction stage, all 44 overlap frequency bins for all beam bands local to a single Hammerhead can be copied into SRAM. With nine Hammerheads performing computations, in the worst case, a Hammerhead will have nine beam bands.

On-Chip Memory Usage. The 44 overlap frequency bins for nine beam bands will occupy 3 KB. The 22 aliased frequency bins for nine beam bands will occupy 2 KB.

Memory Access: DRAM to SRAM. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the overlap frequency bins from DRAM is

$$3{,}168 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 18 \times 10^{-6} \text{ s} \tag{48}$$

Alias Correction. Alias correction for 44 overlap frequency bins and nine beam bands will require 396 flop. The measured sustained throughput of the Hammerhead on matrix operations is 38 Mflop/s. At this rate, the time to perform alias correction is

$$396 \text{ flop} \div \frac{38 \times 10^{6} \text{ flop}}{\text{second}} = 10 \times 10^{-6} \text{ s} \tag{49}$$

Memory Access: SRAM to DRAM. After alias correction, there are 22 aliased frequency bins for each beam band. The alias corrected bins will occupy 2 KB. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the aliased bins to DRAM is

$$1{,}584 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 7 \times 10^{-6} \text{ s} \tag{50}$$

Simultaneous Operation of Three Hammerheads in a Cluster. Because the computation time for alias correction is more than one-third of the SRAM to DRAM memory access time, this memory access cannot be hidden in the computation time, and the execution time for alias correction will be dictated by the total memory access time (see Figure 19). The sum of the memory access times for three Hammerheads is

$$3 \times [(18 \times 10^{-6} \text{ s}) + (7 \times 10^{-6} \text{ s})] = 74 \times 10^{-6} \text{ s} \tag{51}$$

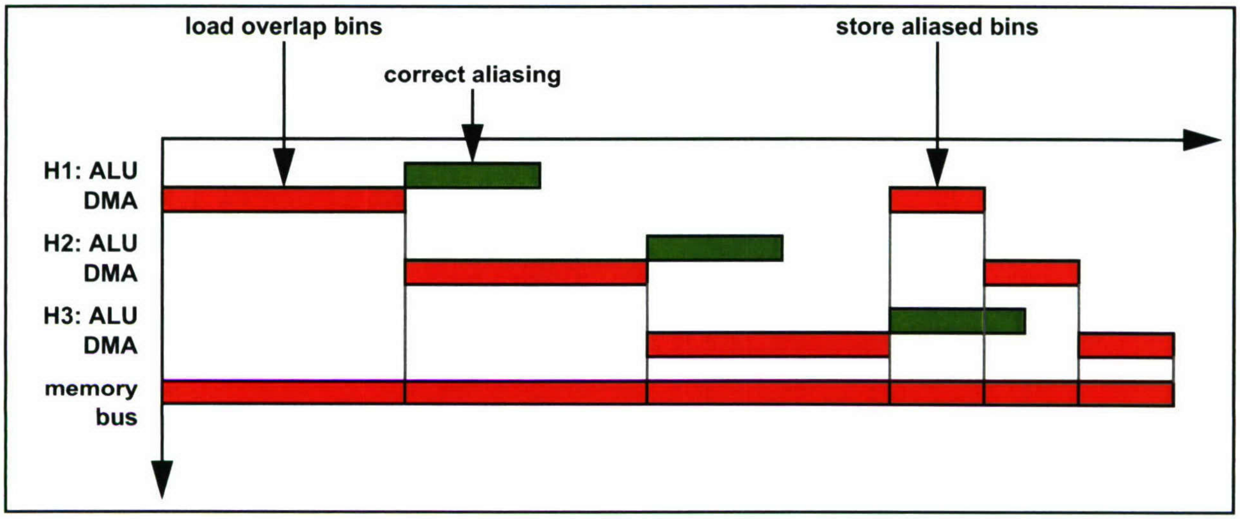

Figure 19: Alias correction execution timing

6.1.8 Overlap and Save Filtering IFFT

For the IFFT portion of overlap and save filtering, we will employ two buffers, as was done with the FFT portion of overlap and save filtering, to allow simultaneous processing and memory access.

On-Chip Memory Usage. Two buffers each capable of storing a complex 256-point vector will occupy 4 KB. The IFFT weights vector will require an additional 1 KB.

Memory Access: DRAM to SRAM. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy a complex 256-point vector from DRAM is

$$2{,}048 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 12 \times 10^{-6} \text{ s} \tag{52}$$

IFFT Processing. A single 256-point complex FFT will require 10,240 flop. The measured sustained throughput of the Hammerhead on FFTs is 293 Mflop/s. At this rate, the time to perform one 256-point complex IFFT is

$$10{,}240 \text{ flop} \div \frac{293 \times 10^{6} \text{ flop}}{\text{second}} = 35 \times 10^{-6} \text{ s} \tag{53}$$

Memory Access: SRAM to DRAM. After the IFFT, we are only interested in the 201 non-overlap time samples, which will occupy 2 KB. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the complex 201-point vector to DRAM is

$$1{,}608 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 7 \times 10^{-6} \text{ s} \tag{54}$$

Simultaneous Operation of Three Hammerheads in a Cluster. Because the computation time for the IFFT stage does not exceed the total memory access time, the memory accesses cannot be completely overlapped with the computations (see Figure 20).

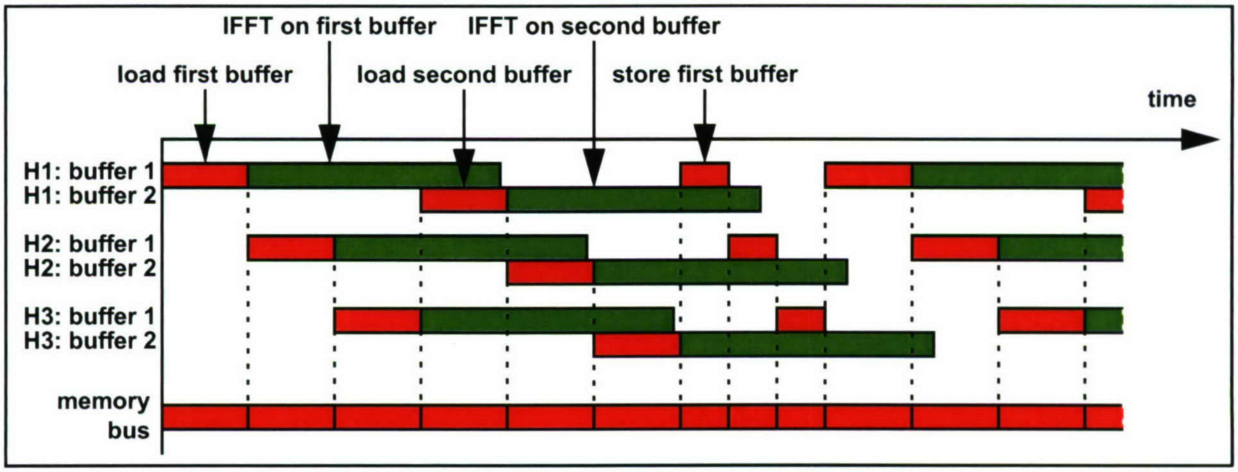

Figure 20: Execution timing for overlap and save IFFT

The total execution time will be dictated largely by the time needed for all the memory accesses:

- three copies for each of eight beam bands from DRAM to SRAM: 279×10^{-6} s
- three copies for each of eight beam bands from SRAM to DRAM: 161×10^{-6} s
- three copies for the ninth beam band from DRAM to SRAM: 35×10^{-6} s
- one 256-point complex FFT: 35×10^{-6} s
- one copy for the ninth beam band from SRAM to DRAM: 7×10^{-6} s

for a total of 517×10^{-6} s.

6.1.9 Phase Correction

For the phase correction stage, all 201 time samples for all beam bands local to a Hammerhead can fit in the on-chip SRAM. With nine Hammerheads performing computations, in the worst case, a Hammerhead will have nine beam bands.

On-Chip Memory Usage. The 201 time samples for nine beams will occupy 14 KB. Because phase correction can be done in-place, we will not need separate input and output buffers.

Memory Access: DRAM to SRAM. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the 201 time samples for nine beam bands from DRAM is

$$14{,}472 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 82 \times 10^{-6} \text{ s} \tag{55}$$

Phase Correction. Phase correction for 201 time samples and nine beam bands will require 10,854 flop. The measured sustained throughput of the Hammerhead on matrix operations of this size is 38 Mflop/s. At this rate, the time to perform phase correction is

$$10{,}854 \text{ flop} \div \frac{38 \times 10^{6} \text{ flop}}{\text{second}} = 283 \times 10^{-6} \text{ s} \tag{56}$$

Memory Access: SRAM to DRAM. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the 201 phase corrected time samples for nine beam bands to DRAM is

$$14{,}472 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 60 \times 10^{-6} \text{ s} \tag{57}$$

Simultaneous Operation of Three Hammerheads in a Cluster. Each of the three Hammerheads performing signal processing in a single Hammerhead cluster will be performing phase correction simultaneously. However, because they share the cluster memory bus, they must access DRAM sequentially. Therefore, the second Hammerhead cannot begin loading its data from DRAM into its first buffer in SRAM until the first Hammerhead has finished loading its data (see Figure 21).

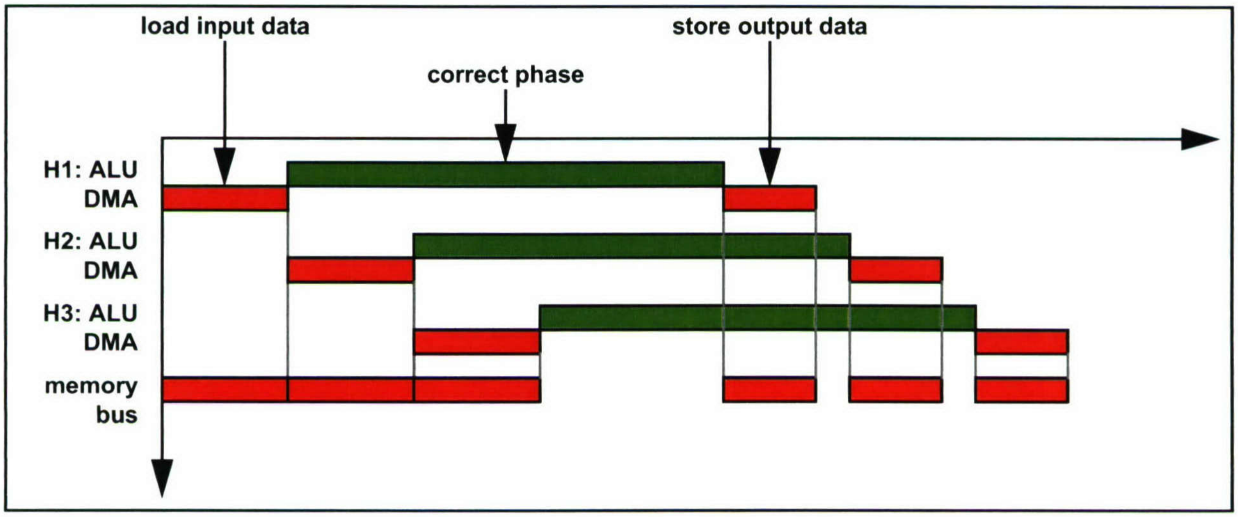

Figure 21: Sequential access of DRAM during phase correction

As the last processor to complete its processing will be the third Hammerhead, we can now estimate the total execution time for this stage:

- wait for the first Hammerhead to load its input data: 82×10^{-6} s
- wait for the second Hammerhead to load its input data: 82×10^{-6} s
- load the input data: 82×10^{-6} s
- correct phase: 283×10^{-6} s
- store the output data: 60×10^{-6} s

for a total of 590×10^{-6} s.

6.1.10 Data Output

Each Hammerhead cluster has, in the worst case, 26 beam bands worth of data. Therefore, the total quantity of data to be transferred to the output FPGA from a single Hammerhead cluster is 42 KB. The measured sustained bandwidth of the link port is 32 MB/s. At this rate, the time to copy the data to the output FPGA is

$$41{,}808 \text{ bytes} \div \frac{32 \times 10^{6} \text{ bytes}}{\text{second}} = 1{,}307 \times 10^{-6} \text{ s} \qquad (58)$$

6.1.11 Timing Summary

The estimated execution time for the various stages are summarized below in Table 8.

Table 8: Estimated Execution Time (Original Algorithm Mode)

Stage	Estimated Execution Time
Data Input	1.9 ms
Overlap and Save: FFT	5.7 ms
Beamforming/Filtering	29.7 ms
Alias Correction	0.1 ms
Overlap and Save: IFFT	0.5 ms
Phase Correction	0.6 ms
Data Output	1.3 ms
Total	**39.8 ms**

Those computational stages where the execution time is determined by the memory access times are those stages where there are relatively few floating-point operations per byte of data accessed. In the overlap and save FFT stage, where the computation time determined the overall execution time, there were 266,240 flop and 41,664 bytes accessed per FFT, for a ratio of nearly six and a half flop per byte. In the alias correction stage, where the memory access time determined the overall execution time, there were 396 flop and 4,752 bytes accessed per Hammerhead, for a ratio of less than a tenth of a flop per byte. Given that the ratio of the peak aggregate computational throughput of the three Hammerheads to the peak bandwidth of the shared memory bus is at least three flop/s per byte/s and as high as six flop/s per byte/s, memory access time will continue to drive the execution time.

6.2 DEFAULT ALGORITHM MODE

6.2.1 Data Input

In the default algorithm mode, each data block has 2,048 samples for each channel. With as many as 18 channels of data being brought into a cluster, the total amount of data per data block is 147 KB.

As was done in the original algorithm mode, the input Hammerhead will accumulate several samples before writing them all to DRAM. In the default mode, we will establish two buffers, each capable of storing 1/4th of the sample extent, or 512 samples, for all 18 channels. Each buffer will therefore be 37 KB in size.

The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy $1/4\text{th}$ of the data into DRAM is

$$36{,}864 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 154 \times 10^{-6} \text{ s} \qquad (59)$$

One such copy to DRAM will occur every 512 samples, or 5.12×10^{-3} s. The total time spent copying data into DRAM is 614×10^{-6} s per data block.

6.2.2 Overlap and Save Filtering FFT

For each channel's worth of data, a 4K-point real FFT must be performed as the first step in the overlap and save filtering.

On-Chip Memory Usage. The 2,048 samples of new data plus the 2,048 samples of old overlap data occupy 16 KB. Even though only 16 KB are copied from DRAM to SRAM for a single vector, the input data vector will occupy twice that space - 33 KB - because the vector must be large enough to store the complex results of the in-place FFT.

To keep the processor busy performing FFTs, we will set up two separate buffers, each 33 KB in size. While the processor is performing an FFT on the data in one buffer, the other buffer will either be filled with more data from DRAM or copied back into DRAM. These two buffers together will occupy 66 KB of the on-chip SRAM. We will also need to store in SRAM the FFT weights, which will occupy 16 KB.

Memory Access: DRAM to SRAM. The 16 KB of input data must be copied from DRAM to SRAM over the shared memory bus. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the input data from DRAM is

$$16{,}384 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 93 \times 10^{-6} \text{ s} \qquad (60)$$

FFT Processing. A single 4K-point real FFT requires 122,880 flop. The measured sustained throughput of the Hammerhead on a real 4K FFT is 307 Mflop/s. At this rate, the time for a single 4K-point real FFT is

$$122{,}880 \text{ flop} \div \frac{307 \ \times 10^{6} \text{ flop}}{\text{second}} = 400 \times 10^{-6} \text{ s} \qquad (61)$$

Memory Access: SRAM to DRAM. After the 4K-point FFT, we are only interested in 140 frequency bins for one subband, which will occupy 1 KB. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the frequency-domain data to DRAM is

$$1{,}120 \text{ bytes} \div \frac{240 \times 10^6 \text{ bytes}}{\text{second}} = 5 \times 10^{-6} \text{ s} \tag{62}$$

Simultaneous Operation of Three Hammerheads in a Cluster. The restrictions on the use of the shared memory bus and the overlapping of computation and memory access that existed for the original algorithm mode also apply to the default algorithm mode. We can estimate the total execution time for this stage, which will be dictated by the third Hammerhead in the first cluster:

- wait for the first Hammerhead to load a 4K-point real vector: 93×10^{-6} s
- wait for the second Hammerhead to load a 4K-point real vector: 93×10^{-6} s
- load a 4K-point real vector: 93×10^{-6} s
- perform 4K-point real FFTs for six channels: $2{,}400 \times 10^{-6}$ s
- store one 140-point complex vector: 5×10^{-6} s

for a total of $2{,}684 \times 10^{-6}$ s .

6.2.3 Beamforming and Filtering: Step 1

Each Hammerhead forms partial beamforming sums from five or six channels' worth of data, then redistributes the data prior to completing the beamforming operation. The mapping of the default algorithm mode onto the DR is the same as the mapping for the original algorithm mode.

On-Chip Memory Usage. The element-space data for six channels and 140 frequency bins will occupy 7 KB. The beamforming/filtering weights vectors for two channels (we use one weights vector while loading the next weights vector) occupy 2 KB. The partial beamforming sum vector will require 1 KB. Together, these data products will require 10 KB.

Memory Access: DRAM to SRAM. The element-space data for six channels and 140 frequency bins will occupy 7 KB. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the element-space data from DRAM is

$$6{,}720 \text{ bytes} \div \frac{176 \times 10^6 \text{ bytes}}{\text{second}} = 38 \times 10^{-6} \text{ s} \tag{63}$$

The beamforming/filtering weights for one channel will occupy 1 KB. The time to copy the weights from DRAM is

$$1{,}120 \text{ bytes} \div \frac{176 \times 10^6 \text{ bytes}}{\text{second}} = 6 \times 10^{-6} \text{ s} \tag{64}$$

Beamforming. Adding the partial beamforming sums from one channel for all 140 frequency bins will require 1,120 flop. The measured sustained throughput of the Hammerhead on a multiply-accumulate operation of this size is 122 Mflop/s. At this rate, the time for this multiply-accumulate operation is

$$1{,}120 \text{ flop} \div \frac{122 \times 10^{6} \text{ bytes}}{\text{second}} = 9 \times 10^{-6} \text{ s} \tag{65}$$

Memory Access: SRAM to DRAM. The partial beamforming sum vector will occupy 1 KB. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the partial sum vector to DRAM is

$$1{,}120 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 5 \times 10^{-6} \text{ s} \tag{66}$$

Simultaneous Operation of Three Hammerheads in a Cluster. The restrictions on the use of the shared memory bus and the overlapping of computation and memory access that existed for the original algorithm mode also apply to the default algorithm mode. We can estimate the total execution time for this stage, which will be dictated by the third Hammerhead in the first cluster:

- wait for the first Hammerhead to load the element-space data for six channels: 38×10^{-6} s
- wait for the second Hammerhead to load the element-space data for six channels: 38×10^{-6} s
- load the element-space data for six channels: 38×10^{-6} s
- for each beam band:
  - load the beamforming/filtering weights for all six channels for all three Hammerheads: $6 \times 10^{-6} \text{ s} \times 3 \times 6 = 115 \times 10^{-6} \text{ s}$
  - perform multiply/accumulate for the sixth channel: 9×10^{-6} s
  - store partial beamforming sum in DRAM: 5×10^{-6} s

for a total of $1{,}912 \times 10^{-6}$ s for 14 beam bands.

6.2.4 Beamforming/Filter Application Step 2

In the second step of beamforming/filtering, we accumulate the three partial sums that were computed within a single Hammerhead cluster. Specifically, each Hammerhead loads from DRAM all three partial sums for one-third of the $N_{beams} \times N_{subbands}$ beam bands, where each input partial sum is computed from six channels' worth of element-space data, and combines them to form one set of output partial sums for one-third of the beam bands.

Before we load the three partial sums computed during the first step of beamforming/filtering, we need to synchronize the three Hammerheads in each cluster to force the processors to wait for the previous step to complete.

On-Chip Memory Usage. There is sufficient space in the on-chip SRAM to allow all 140 frequency bins to be processed in one batch. One-third of 14 beam bands, rounded up, is five beam bands. The three sets of input partial sums for five beam bands and 140 frequency bins will occupy 17 KB. The output partial sums for five beam bands and 140 frequency bins will occupy 6 KB. Together, these data products will require 22 KB.

Memory Access: DRAM to SRAM. The three sets of input partial sums will occupy 17 KB. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the input partial sums from DRAM is

$$16{,}800 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 95 \times 10^{-6} \text{ s} \tag{67}$$

Beamforming. Forming the cluster-wide partial sums for five beam bands and 140 frequency bins will require 2,800 flop. The measured sustained throughput of the Hammerhead on matrix operations of this size is 38 Mflop/s. At this rate, the time for the second step of beamforming and filter application is

$$2{,}800 \text{ flop} \div \frac{38 \times 10^{6} \text{ flop}}{\text{second}} = 73 \times 10^{-6} \text{ s} \tag{68}$$

Memory Access: SRAM to DRAM. The cluster-wide partial sums for five beam bands and 140 frequency bins will occupy 6 KB. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the partial sums to DRAM is

$$5{,}600 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 23 \times 10^{-6} \text{ s} \tag{69}$$

Simultaneous Operation of Three Hammerheads in a Cluster. The restrictions on the use of the shared memory bus and the overlapping of computation and memory access that existed for the original algorithm mode also apply to the default algorithm mode. We can estimate the total execution time for this stage, which will be dictated by the third Hammerhead in the first cluster:

- wait for the first Hammerhead to load its three sets of input partial sums: 95×10^{-6} s
- wait for the second Hammerhead to load its three sets of input partial sums: 95×10^{-6} s
- load the three sets of input partial sums: 95×10^{-6} s
- add the three partial sums: 73×10^{-6} s
- store the output partial sum in DRAM: 23×10^{-6} s

for a total of 383×10^{-6} s for all 140 frequency bins.

6.2.5 Beamforming and Filtering: Inter-Cluster Communication

Prior to the final step of beamforming and filter application, inter-cluster communication is necessary to exchange beamforming partial sums between clusters. At the end of the beamforming and filter application stage, we would like to have the beams equally distributed over the three Hammerhead clusters, with the entire frequency subband in a single Hammerhead cluster. This arrangement will allow for the remaining stages of the BSAR processing to be done local to a single Hammerhead.

Before we transfer the partial sums between the Hammerhead clusters, we need to synchronize all nine Hammerheads performing signal processing to force the processors to wait for the previous step to complete.

Prior to the inter-cluster communication, each cluster has a partial sum for all 14 beam bands and 140 frequency bins. After the inter-cluster communication, each cluster will have all three partial sums for a third of the 14 beam bands. This communication operation entails sending, in the worst case, five beam bands of partial sums to one Hammerhead cluster and five beam bands of partial sums to the other Hammerhead cluster. This operation can be performed in two stages: data transfer to the neighboring cluster to the left, and data transfer to the neighboring cluster to the right. In both directions, the worst case quantity of data to be transferred is 11 KB for five partial sums and 140 frequency bins.

On-Chip Memory Usage. There is sufficient space in the SRAM to allow all 140 frequency bins to be transferred at one time. The two outgoing sets of partial sums will require 11 KB. The two incoming sets of partial sums will require an additional 11 KB, for a total of 22 KB.

Memory Access: DRAM to SRAM. The two sets of outgoing partial sums will occupy 11 KB. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the outgoing data from DRAM is

$$11{,}200 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 64 \times 10^{-6} \text{ s} \tag{70}$$

Inter-Cluster Communication. The measured sustained bandwidth of the link ports is 32 MB/s. At this rate, the time to perform these two transfers is

$$11{,}200 \text{ bytes} \div \frac{32 \times 10^{6} \text{ bytes}}{\text{second}} = 350 \times 10^{-6} \text{ s} \tag{71}$$

Memory Access: SRAM to DRAM. The two sets of incoming partial sums will occupy 11 KB. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy both sets of partial sums to DRAM is

$$11{,}200 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 47 \times 10^{-6} \text{ s} \tag{72}$$

The total time needed to transfer all 140 frequency bins is

$$(64 + 350 + 47) \times 10^{-6}\ \text{s} = 460 \times 10^{-6}\ \text{s} \tag{73}$$

6.2.6 Beamforming/Filter Application Step 3

In the third step of beamforming/filtering, we combine the partial sums across the three Hammerhead clusters. Specifically, each of the nine Hammerheads (three Hammerheads in each of the three clusters) performing signal processing will combine the three partial sums for one-ninth of the beam bands to produce the total beamformed sum.

Before we load the three partial sums computed during the first two step of beamforming/filtering, we need to synchronize all nine Hammerheads performing signal processing to force the processors to wait for the previous step to complete.

On-Chip Memory Usage. One-ninth of the 14 beam bands, rounded up, is two beam bands. The three sets of partial sums for two beam bands and 140 frequency bins occupy 7 KB. The output total sums for two beam bands and 140 frequency bins occupy 2 KB. Together, these data products require 9 KB.

Memory Access: DRAM to SRAM. The three sets of partial sums will occupy 7 KB. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the partial sums from DRAM is

$$6{,}720\ \text{bytes} \div \frac{176 \times 10^{6}\ \text{bytes}}{\text{second}} = 38 \times 10^{-6}\ \text{s} \tag{74}$$

Beamforming. Forming the beamformed output for two beam bands and 140 frequency bins will require 1,120 flop. The measured sustained throughput of the Hammerhead on matrix operations of this size is 38 Mflop/s. At this rate, the time for the third step of beamforming and filter application is

$$1{,}120\ \text{flop} \div \frac{38 \times 10^{6}\ \text{flop}}{\text{second}} = 29 \times 10^{-6}\ \text{s} \tag{75}$$

Memory Access: SRAM to DRAM. The beamformed output for 140 frequency bins will occupy 2 KB of memory. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the beamformed output back to DRAM is

$$2{,}240\ \text{bytes} \div \frac{240 \times 10^{6}\ \text{bytes}}{\text{second}} = 9 \times 10^{-6}\ \text{s} \tag{76}$$

Simultaneous Operation of Three Hammerheads in a Cluster. The restrictions on the use of the shared memory bus and the overlapping of computation and memory access that existed for the original algorithm mode also apply to the default algorithm mode. We can estimate the total execution time for this stage, which will be dictated by the third Hammerhead in the first cluster:

- wait for the first Hammerhead to load its three sets of partial sums: 38×10^{-6} s
- wait for the second Hammerhead to load its three sets of partial sums: 38×10^{-6} s
- load the three sets of partial sums: 38×10^{-6} s
- add the three partial sums: 29×10^{-6} s
- store the output total sum: 9×10^{-6} s

for a total of 153×10^{-6} s for all 140 frequency bins.

We can now calculate the projected execution time for the entire beamforming/filtering stage:

- step 1: $1{,}912 \times 10^{-6}$ s
- step 2: 383×10^{-6} s
- inter-cluster communication: 460×10^{-6} s
- step 3: 153×10^{-6} s

for a total of $2{,}908 \times 10^{-6}$ s.

6.2.7 Alias Correction

For the alias correction stage, all 24 overlap frequency bins for all beam bands local to a single Hammerhead can be copied into SRAM. With nine Hammerheads performing computations, in the worst case, a Hammerhead will have two beam bands.

On-Chip Memory Usage. The 24 overlap frequency bins for two beam bands will occupy 384 bytes. The 12 aliased frequency bins for nine beam bands will occupy 192 bytes.

Memory Access: DRAM to SRAM. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the overlap frequency bins from DRAM is

$$384 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 2 \times 10^{-6} \text{ s} \tag{77}$$

Alias Correction. Alias correction for 24 overlap frequency bins and two beam bands will require 48 flop. The measured sustained throughput of the Hammerhead on matrix operations is 38 Mflop/s. At this rate, the time to perform alias correction is

$$48 \text{ flop} \div \frac{38 \times 10^{6} \text{ flop}}{\text{second}} = 1 \times 10^{-6} \text{ s} \tag{78}$$

Memory Access: SRAM to DRAM. After alias correction, there are 12 aliased frequency bins for each beam band. The alias corrected bins will occupy 192 bytes. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the aliased bins to DRAM is

$$192 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 1 \times 10^{-6} \text{ s} \tag{79}$$

Simultaneous Operation of Three Hammerheads in a Cluster. Because the computation time for alias correction is more than one-third of the SRAM to DRAM memory access time, this memory access cannot be hidden in the computation time, and the execution time for alias correction will be dictated by the total memory access time. The sum of the memory access times for three Hammerheads is

$$3 \times [(2 \times 10^{-6} \text{ s}) + (1 \times 10^{-6} \text{ s})] = 9 \times 10^{-6} \text{ s} \tag{80}$$

6.2.8 Overlap and Save Filtering IFFT

For the IFFT portion of overlap and save filtering, we will employ two buffers, as was done with the FFT portion of overlap and save filtering, to allow simultaneous processing and memory access.

On-Chip Memory Usage. Two buffers each capable of storing a complex 128-point vector occupy 2 KB. The IFFT weights vector will require an additional 512 bytes.

Memory Access: DRAM to SRAM. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy a complex 128-point vector from DRAM is

$$1{,}024 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 6 \times 10^{-6} \text{ s} \tag{81}$$

IFFT Processing. A single 128-point complex FFT will require 4,480 flop. The measured sustained throughput of the Hammerhead on FFTs is 250 Mflop/s. At this rate, the time to perform one 128-point complex IFFT is

$$4{,}480 \text{ flop} \div \frac{250 \times 10^{6} \text{ flop}}{\text{second}} = 18 \times 10^{-6} \text{ s} \tag{82}$$

Memory Access: SRAM to DRAM. After the IFFT, we are only interested in the 64 non-overlap time samples, which will occupy 512 bytes. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the complex 64-point vector to DRAM is

$$512 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 2 \times 10^{-6} \text{ s} \tag{83}$$

Simultaneous Operation of Three Hammerheads in a Cluster. The restrictions on the use of the shared memory bus and the overlapping of computation and memory access that existed for the original algorithm mode also apply to the default algorithm mode. We can estimate the total execution time for this stage, which will be dictated by the third Hammerhead in the first cluster:

- three copies for the first beam band from DRAM to SRAM: 17×10^{-6} s
- three copies for the first beam band from SRAM to DRAM: 6×10^{-6} s
- three copies for the second beam band from DRAM to SRAM: 17×10^{-6} s
- one 128-point complex FFT: 18×10^{-6} s
- one copy for the second beam band from SRAM to DRAM: 2×10^{-6} s

for a total of 61×10^{-6} s.

6.2.9 Phase Correction

For the phase correction stage, all 64 time samples for all beam bands local to a Hammerhead can fit in the on-chip SRAM. With nine Hammerheads performing computations, in the worst case, a Hammerhead will have two beam bands.

On-Chip Memory Usage. The 64 time samples for two beams will occupy 1 KB. Because phase correction can be done in-place, we will not need separate input and output buffers.

Memory Access: DRAM to SRAM. The measured sustained bandwidth of the shared memory bus for reads from DRAM is 176 MB/s. At this rate, the time to copy the 64 time samples for nine beam bands from DRAM is

$$1{,}024 \text{ bytes} \div \frac{176 \times 10^{6} \text{ bytes}}{\text{second}} = 6 \times 10^{-6} \text{ s} \tag{84}$$

Phase Correction. Phase correction for 64 time samples and two beam bands will require 768 flop. The measured sustained throughput of the Hammerhead on matrix operations of this size is 38 Mflop/s. At this rate, the time to perform phase correction is

$$768 \text{ flop} \div \frac{38 \times 10^{6} \text{ flop}}{\text{second}} = 20 \times 10^{-6} \text{ s} \tag{85}$$

Memory Access: SRAM to DRAM. The measured sustained bandwidth of the shared memory bus for writes to DRAM is 240 MB/s. At this rate, the time to copy the 64 phase corrected time samples for nine beam bands to DRAM is

$$1{,}024 \text{ bytes} \div \frac{240 \times 10^{6} \text{ bytes}}{\text{second}} = 4 \times 10^{-6} \text{ s} \tag{86}$$

Simultaneous Operation of Three Hammerheads in a Cluster. The restrictions on the use of the shared memory bus and the overlapping of computation and memory access that existed for the original algorithm mode also apply to the default algorithm mode. We can estimate the total execution time for this stage, which will be dictated by the third Hammerhead in the first cluster:

- wait for the first Hammerhead to load its input data: 6×10^{-6} s
- wait for the second Hammerhead to load its input data: 6×10^{-6} s
- load the input data: 6×10^{-6} s
- correct phase: 20×10^{-6} s
- store the output data: 4×10^{-6} s

for a total of 42×10^{-6} s.

6.2.10 Data Output

Each Hammerhead cluster has, in the worst case, five beam bands worth of data. Therefore, the total quantity of data to be transferred to the output FPGA from a single Hammerhead cluster is 3 KB. The measured sustained bandwidth of the link port is 32 MB/s. At this rate, the time to copy the data to the output FPGA is

$$2{,}560 \text{ bytes} \div \frac{32 \times 10^{6} \text{ bytes}}{\text{second}} = 80 \times 10^{-6} \text{ s} \tag{87}$$

6.2.11 Timing Summary

The estimated execution time for the various stages are summarized below in Table 8.

Table 9: Estimated Execution Time (Default Algorithm Mode)

Stage	**Estimated Execution Time**
Data Input	0.6 ms
Overlap and Save: FFT	2.7 ms
Beamforming/Filtering	2.9 ms
Alias Correction	0.0 ms
Overlap and Save: IFFT	0.1 ms
Phase Correction	0.0 ms
Data Output	0.1 ms
Total	**6.4 ms**

7. SUMMARY AND CONCLUSIONS

In this report, we have described both the BSAR DR frequency-domain algorithm and the 12-Hammerhead DR architecture. Given these two components, we have proposed a mapping of the algorithm onto the architecture, and have estimated the execution time for both the original algorithm mode and the new default mode using measurements of the computation and communication performance.

Given the proposed mapping and these efficiency estimates, we project that execution of the DR algorithm in the original mode will take approximately 39.8 ms, which, after considering the need for 50% processor spare, is within the 42.9 ms available (the 39.8 ms estimated execution time leaves 62% spare processor capacity). The algorithm in the default mode will take approximately 6.4 ms, which is within the 13.7 ms available (the 6.4 ms estimated execution time leaves 220% spare processor capacity). A key observation from this mapping analysis is that the memory access time will heavily dominate the overall execution timing and, given the likely contention issues, will therefore be a risk area. Alternate mappings that make greater use of the link port connections between Hammerheads to alleviate some of the load on the shared memory bus were analyzed in the original version of this report and were found to have worse performance.

Our analysis shows that the BSAR DR will be able to perform its signal processing within the available time, although it is possible that some of the processor margin will be consumed if shared memory bus contention is worse than we anticipate. Also, we expect that the DR will have sufficient SRAM and DRAM to store input, intermediate, and output data, although, with the original algorithm parameters, some stages will consume significant portions of the memory margin.

LIST OF ACRONYMS

ADC	Analog to Digital Converter
ADCAP	ADvanced CAPability
AGC	Automatic Gain Control
ALU	Arithmetic and Logic Unit
BES	Broadband Evaluation System
BSAR	Broadband Sonar Analog Receiver
CBASS	Common Broadband Advanced Sonar System
DMA	Direct Memory Access
DR	Digital Receiver
DRAM	Dynamic Random-Access Memory
DSP	Digital Signal Processor
ENOBs	Effective Number Of Bits
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
flop	FLoating-point OPeration, a measure of processing workload
flop/s	FLoating-point OPerations per second, a measure of processing throughput
FPGA	Field-Programmable Gate Array
GCB	Guidance and Control Box
IFFT	Inverse Fast Fourier Transform
KB	Kilo Byte, or 1,000 bytes
MB	Mega Byte, or 1,000,000 bytes
NUWC	the Naval Undersea Warfare Center
PROM	Programmable Read-Only Memory
SIMD	Single Instruction stream, Multiple Data streams
SDRAM	Synchronous Dynamic Random-Access Memory
SNR	Signal to Noise Ratio
SRAM	Static Random-Access Memory

REFERENCES

[1] J. Proakis and D. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd edition, Upper Saddle River, New Jersey: Prentice Hall (1996).

[2] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, Philadelphia: Society for Industrial and Applied Mathematics (1992).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 5 February 2002	**2. REPORT TYPE** Project Report	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE

BSAR Computational Analysis and Proposed Mapping, Revision 1

5a. CONTRACT NUMBER
FA8721-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

M. Arakawa

5d. PROJECT NUMBER
1048

5e. TASK NUMBER
1

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420-9108

8. PERFORMING ORGANIZATION REPORT NUMBER

Project Report CBASS-1, Revision 1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Capt. David A. Portner, PMS404E4
Naval Sea Systems Command
1333 Isaac Hull Ave., S.E.
Stop 7015
Washington Navy Yard, DC 20376-7015

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
ESC-TR-2006-072

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report details the MIT Lincoln Laboratory computational performance analysis of the CBASS BSAR. We describe:

- the BSAR algorithm, in both the original mode and the new, default mode
- the algorithm's computational workload for both modes
- the 12-Hammerhead DR on which the BSAR algorithm is executed
- a proposed mapping of the algorithm onto the DR
- an estimated execution time for the algorithm in both modes using the proposed mapping
- a memory usage analysis

Our analysis indicates that the DR will be able to handle the BSAR algorithm in both the original and the new, default modes with sufficient spare processor capacity. Furthermore, there is sufficient memory for the various input, intermediate, and output data products, although some of the memory margin will be consumed in the original algorithm mode.

15. SUBJECT TERMS

torpedo digital signal processing
signal processor architecture
computational throughput
algorithm mapping

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	None	74	**19b. TELEPHONE NUMBER** *(include area code)*